AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

REVISTA MENSAL Nº10 ANO 1 MARÇO 1989 350$00

DOS 4.0

O MS-DOS POR DENTRO

LOCOTEXTO A DUAS COLUNAS

A PERSPECTIVA DA...

QUEM ~~TRAMOU~~ JOGOU O ROGER RABBIT?

EDITORIAL

REVISTA MENSAL
Nº 10 ANO 1
MARÇO 1989
PREÇO 350$00

PROPRIEDADE:
PUBLINFOR, Publicações e Comércio de Artigos de Informática, S.A. —

DIRECÇÃO:
Fernando Prata

COLABORADORES:
Eng. Mário Leite, Dr. Maria de Lurdes Leite, António Torres Martins, António Cardoso, Paulo Pinheiro

PRODUÇÃO GRÁFICA:
GRAFICRIA, Publicidade e Artes Gráficas, Lda. — Rua Alfredo Roque Gameiro, 21-1º Dtº Tel: 76 27 32

PUBLICIDADE E ASSINATURAS:
PUBLINFOR
Centro de Escritórios das Laranjeiras — Urbanização das Laranjeiras — Praça Nuno Rodrigues dos Santos, 7-2º Piso - Sala 13 - 1600 LISBOA
Telf: 7269011 Telex 62752 Simose P
Fax: 7269985

TIRAGEM: 11500 exemplares

PREÇO DE CAPA: 350$00

DISTRIBUIÇÃO:
ELECTROLIBER

— Nº PES. COLECT. 502009870
— Nº REG. D.G.C.S. 112959
— DEPÓSITO LEGAL Nº 20669/88

QUERIA UM JOGO PARA O "MÍUDO"

Esta é apenas uma das muitas desculpas que os "adultos" utilizam, quando adquirem nas lojas os jogos para o computador que possuiem em casa. Com efeito, muitos destes "adultos" continuam a pensar que os jogos de computador são apenas para os "míudos", e que se colocariam numa posição rídicula se afirmassem em público que gostam de jogar, esses mesmos "jogos de míudos". A realidade é, contudo, um pouco diferente, e ridículos estão a ser todos os "adultos" que gostando de jogar um qualquer jogo não o querem admitir.

Um dia, ao ler meia dúzia de linhas escritas por alguém cujo nome não fixei, a ideia fundamental para a explicação deste "fenómeno" foi-me transmitida. Assim, segundo essa celebridade desconhecida, um adulto define-se facilmente como aquele que mantendo em si uma parte de criança a consegue separar racionalmente de toda a sua experiência de vida posterior. Deste modo, todos os adultos são simultâneamente crianças, e todos aqueles que negarem a sua parte de criança nunca poderão considerar-se adultos. É claro que isto é uma ideia nova para todas as pessoas definidas pelos primeiros exemplos apontados, o que não é de admirar visto que todos eles têm ainda muito que aprender até se tornarem adultos "a sério" (independentemente da idade que possam possuir).

Neste ponto do discurso, e tentando juntar mais achas à fogueira, colocar-se-ía a dúvida de saber se no fundo é mais importante ser-se adulto, ou ser-se criança. E com isto passariamos horas a discutir "o sexo dos anjos".

Pessoalmente penso que o facto de gostar de jogar um bom jogo de computadores não me rouba minimamente as capacidades de programador, ou de adulto participante de uma sociedade com demasiadas responsabilidades pessoais. Assumo naturalmente a parte de criança que quero continuar a guardar, e tenho pena de todos os que não pensam desta forma.

Talvez por todas estas razões neste número da AM damos um destaque especial a um jogo de computador. Nunca nos consideraremos por isso uma publicação "inferior", pois pensamos que o importante é sermos racionais e não falarmos só de jogos, e aí pensamos que tudo está dito se folhear calmamente o exemplar da AM que agora começou a ler.

O "sumo" habitual está presente em muitas páginas para satisfazer as mais diversas necessidades de informação e conhecimentos por parte dos leitores, mas os jogos também são necessários. Tente, por exemplo, esquecer as trezentas "coisas" que tinha que fazer e não fez ao longo do dia, e perder toda a má disposição e dores de cabeça em frente do Roger Rabbit, rindo "a bandeiras despregadas", e revivendo os momentos do filme, que se calhar nem viu porque andava muito ocupado com as suas "actividades de adulto".

SUMÁRIO

4 NOTICÍAS

6 DOS 4.0

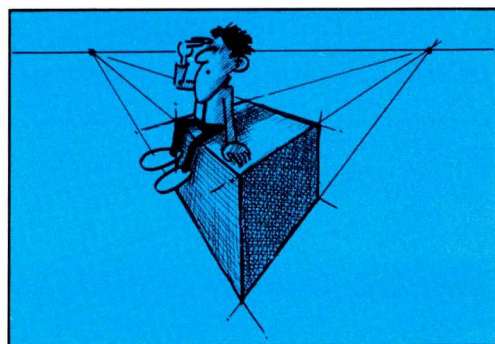
Se entre o MS-DOS 3.3 e a versão imediatamente anterior não se pode dizer que existem grandes diferenças, entre o DOS 4.0 e qualquer uma das versões anteriores quase poderiamos afirmar (exagerando um "pouco"), que as poucas semelhanças existentes são puras coincidências. É claro que, tal como acabámos de referir, o que afirmámos é um exagero, mas as grandes diferenças entre esta nova versão do DOS e todas as outras que a antecederam existem e foi para falar delas que a AM "arrancou" em 4.0 e decidiu publicar este artigo.

11 AUMENTE A SUA LIVRARIA DE C (III E ÚLTIMA PARTE)

Concluindo a ampliação da sua livraria de C, neste número da AM apresentamos as últimas rotinas inseridas no "pacote" de rotinas gráficas. Muitas instruções para teclar, mas bastante utilidade nas listagens apresentadas, e o resultado final será sempre um C "Cualitativamente" superior.

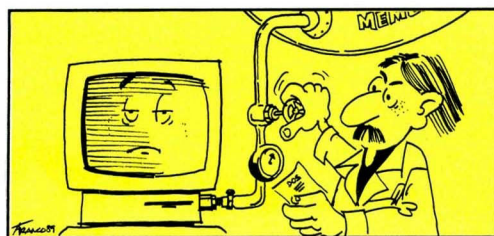
18 O MS-DOS POR DENTRO

O conhecimento do MS-DOS "por fora" é do domínio público, mas, o conhecimento do MS-DOS por dentro é algo que só poderá ter se começar a seguir esta série da artigos aqui iniciada.

22 UMA PERSPECTIVA SOBRE A PERSPECTIVA

A representação de gráficos em computador pode parecer algo de muito complexo, e na realidade até é algo de muito complexo. Contudo, para alguns programadores a representação tridimensional de um objecto num ecrã de computador, tornou-se tão simples como utilizar este poderoso instrumento de trabalho para somar 2 + 2. Siga atentamente as explicações do João Henrique sobre o assunto e, começe desde logo a criar, por exemplo, o seu Space Shutle com traseira tipo Ford T.

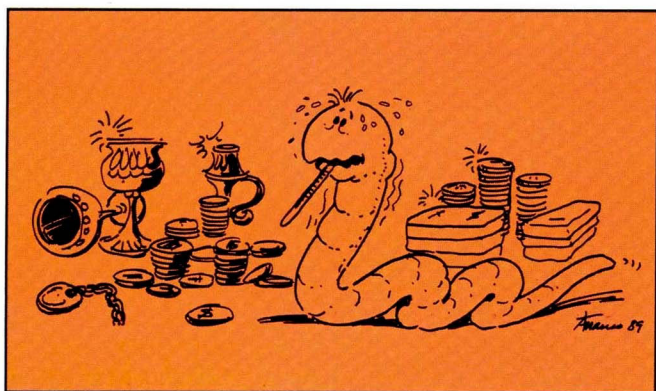
42 WINDOWS 2.03 — NOVA VERSÃO OU APENAS UM "UPGRADE"

Sem pretender ser uma análise exaustiva, esta análise da nova versão do Windows pode dar-lhe algumas indicações sobre as principais diferenças entre as versões antigas e a moderna, proporcionando-lhe ainda algumas comparações entre as versões Windows/8086 e Windows/80386. Se trabalha todos os dias inserido neste ambiente talvez não aprenda aqui nada de novo, no entanto, se esse não é o seu caso, é muito possivel que as linhas que dedicamos a esta aplicação sejam um bom cartão de visita.

45 ATRIBUIÇÃO DE MEMÓRIA NO MS-DOS

Embora os programadores de Assembler 8086/88 não sejam ainda em número suficiente no nosso país para justificar grandes explicações técnicas sobre a matéria em causa, algumas páginas a eles dedicadas serão sempre interessantes já que num futuro próximo poderão interessar todos os outros leitores.

52 QUEM TRAMOU O ROGER RABBIT

Depois do Bugs Bunny, Roger é talvez o coelho mais famoso mundialmente. Tramado, ou não, Roger Rabbit continua a fazer rir inúmeras pessoas nas salas de cinema, brincando num mundo semi-real em que tudo é possivel, e o possivel fica muito próximo de não o ser.
Roger Rabbit, tal como Clark Gable, Fred Astaire, e muitas outras estrelas ficará um dia na história do cinema. Mas... Será que vai ficar apenas por aí?

57 A FEBRE DO OURO

36, 37, 37.5, 38, 40, 42, 43.... A Febre do Ouro!!!

60 UMA "SPRITEADELA" NO CPC 464

O termo "sprite" pode não ser sinónimo de bebida refrescante. Mas a mesma palavra pode ser, tal como será neste artigo dedicado ao CPC, sinónimo de uma "fresca" forma de manipular gráficos no ecrã. Como diriam todos os leitores familiarizados com os "sprites", se uma imagem vale mais do que mil palavras, um "sprite" vale, "à vontade", mais de mil instruções de programação.

64 IMPRESSÃO A DUAS COLUNAS

Formatar os textos escritos com o Locoscript através de uma pequena rotina escrita em BASIC pode ser a solução que esperava para editar o seu boletim de empresa, o seu jornal de escola, ou mesmo escrever os seus trabalhos com um aspecto mais profissional, e menos vulgar em termos de processamento de texto. Curioso em tudo isto, é que a rotina em causa se introduz no computador em poucos minutos, e a operação por ela levada a cabo apenas demora alguns segundos. Conselho AM: ver para crer.

65 PCW 8256 = VT 52 ?

Sabia que o seu PCW pode emular com alguma facilidade um terminal VT 52 ? Se a resposta à questão anterior é afirmativa, então resta-lhe saber como perfazer essa emulação. Nesse caso, todo o tempo que está a perder ao ler esta pequena apresentação do artigo deve ser aproveitado, sem demora, para uma leitura atenta do mesmo.

AMSTRAD NETWORK
Uma solução de elevada performance

Os utilizadores Amstrad têm agora disponível mais uma solução de elevada performance: a Rede Amstrad. Com a Rede Amstrad, é possível ligar vários computadores pessoais que partilham a informação entre si. Os diversos utilizadores individuais têm, como membros da rede, acesso a todos os meios informáticos existentes: hardware, software e impressoras.
A instalação da rede é simples: os cabos e fichas são ligados tão facilmente como uma impressora, unindo todos os computadores participantes na rede. Completa a instalação o AMSNOS (Amstrad Network Operating System). Este software não cria dificuldades ao utilizador porque não gera novas rotinas nem novos ou complexos comandos.
Para o utilizador, operar em rede é tão simples como se se tratasse de um PC isolado. A grande diferença reside no maior e mais diversificado mundo em que vai trabalhar. O acesso e a introdução de dados, numa perspectiva multi-utilizador, a um tratamento de texto ou à contabilidade são muito mais simples e eficientes.

CORREIO ELECTRÓNICO
Novas possibilidades

Com a Rede Amstrad, abrem-se novas possibilidades à circulação de informação no local de trabalho usando o correio electrónico. Mensagens, cartas, memorandos podem ser enviados para um posto de trabalho específico, podem circular em toda a rede ou só num grupo seleccionado.
As informações podem também ser muito apreciadas pelos utilizadores que assim obtêm significativa economia de tempo enquanto são minimizados os problemas de espaço.
Como gestor a Rede Amstrad possibilita o controlo (diário, necessário), através de chaves de acesso, do acesso dos vários utilizadores às diversas informações. Por outro lado, o gestor pode estar sempre por dentro do que se passa nos diversos serviços já que pode ter acesso a toda a informação guardada no sistema, bem como estabelecer ligação a qualquer posto de trabalho.
Uma referência a outra vantagem da Rede Amstrad: o "file server" que pode também ser usado como posto de trabalho já que a memória requerida é pequena.

AMSTRAD NETWORK
Especificações técnicas

1) Rede

Release:	1.2
Sistema operativo:	MS-DOS3.2 ou 3.3
Network hardware:	Omninet
Nº máximo de nódulos por rede:	6
Entradas de directório por drive:	256 k
Computadores:	Amstrad, IBM PC, XT, AT ou compatíveis

2) Uso de Memória

—File Server:

Memória mínima requerida:	512 K
PC/N)* (*):	200 K
Front-end cache:	5 K
Back-end cache:	16 K

(*) Este valor destina-se a um disco rígido de 20 Mb. Para discos com maior capacidade, acrescentar 1 K por Mb do disco partilhado e para os IBM PC/AT ou compatíveis 2 K por Mb do disco partilhado.

—Posto de Trabalho:

Memória mínima requerida:	256 K
PC/NOS:	66 K
Front-end cache:	5 K
Back-end cache:	N/A

3) Requisitos

— File Server: Amstrad ou IBM compatível com pelo menos 512 K (memória RAM); com disco rígico; com monitor monocromático ou policromático; sistema operativo MS-DOS 3.3 ou 3.2.

— Postos de Trabalho: Amstrad ou IBM compatível com 256 K (memória RAM); com disco rígido ou com 1 ou 2 drives; monitor monocromático ou policromático; sistema operativo MS-DOS 3.3 ou 3.2.

ANDEBOL DE SETE
1º TORNEIO INTERNACIONAL AMSTRAD

A equipa do F.C. Porto foi a vencedora do 1º Torneio Internacional "AMSTRAD" do Andebol de Sete, realizado na Maia, nos dias 11 e 12 de Fevereiro.
O 1º Torneio Amstrad contou com a participação de quatro das melhores equipas da Península Ibérica nesta modalidade: F.C. Porto, ABC de Braga, Teka e Laguisa.
Perante numerosa e entusiasta assistência, os quatro jogos deste torneio tiveram os seguintes resultados: ABC, 20 — Laguisa, 19 — F.C. Porto, 21 — Teka, 19.
No jogo de apuramento do 3º e 1º lugares, em aue participaram as duas equipas espanholas, venceu a da Teka por 23—20.
Na esperada final, o F.C. Porto levou de vencida o ABC de Braga pelo tangencial de 19—17.
O 1º Torneio Amstrad saldou-se por um êxito desportivo. Contribuiu não só para divulgar ainda mais o nome da conhecida marca de computadores, mas também "para o desenvolvimento do derporto em Portugal, neste caso o andebol, um desporto bomito e espectacular que cada vez tem mais adeptos no nosso país", sublinhou o Engº Francisco Magalhães, Director-Geral da Sopsi.

AMSTRAD PC 1640
O Computador mais vendido em Inglaterra

O Amstrad PC 1640 foi o microcomputador mais vendido em Inglaterra, no mês de Outubro de 1988, revelou um estudo da "ROMTEC REVIEW".
De acordo com este trabalho, a Amstrad estava em 2º lugar no "ranking" inglês das marcas de microcomputadores, com uma quota de mercado de 18%, logo a seguir à IBM (24%), e deixando atrás de si, por exemplo, a Compaq, a Apple e a Olivetti.

LÍDERES EM VENDAS

1.	**Amstrad PC 1640**	6.	**Apricot Xeni**
2.	**IBM 30**	7.	**IBM 50**
3.	**IBM 50 z**	8.	**Compaq Deskpro 386**
4.	**Compaq Deskpro 286**	9.	**IBM 60**
5.	**Apple MacSE**	10.	**IBM PC-AT**

A Amstrad liderava, contudo, o sector de mercado dos microcomputadores com processador 8086-8, onde detinha uma quota de 47%, relegando para segundo lugar a IBM (16%) e distanciando-se claramente de outras conhecidas marcas: Zenith (7%), Olivetti (5%) e Toshiba (5%).
Nesta altura a Amstrad ainda não tinha começado a comercialização da sua nova série PC 2000, por isso não aparece cotada nos mercados dos computadores com os processadores 80286 e 80386. No entanto, tudo leva a crer que, a muito curto prazo, a Amstrad atinja também aqui uma posição de destaque.
A tendência parece revelar um declínio significativo da IBM, enquanto se assiste a uma quota de mercado (39%) inferior à dos outros dois líderes, Amstrad (17% 8086-8) e Compaq (45% 80386).
O peso relativo dos segmentos do mercado inglês segundo a "ROMTEC REVIEW" apresentam-se assim: 8086-8 com 42% das unidades vendidas (21% em valor); 80386 com 41% das unidades vendidas (48% em valor); 80386 com 17% das unidades vendidas (31% em valor).

ANÁLISE: DOS 4.0

1. INTRODUÇÃO

Há muito que se esperava pela saída de uma nova versão do DOS que resolvesse determinadas questões prementes que hà longo tempo se colocavam para os utilizadores deste sistema operativo. Uma delas era o facto do DOS não conseguir aproveitar mais do que os 640Kb de RAM, o que já frustava bastantes utilizadores que, apesar de possuirem computadores com 1Mb ou mais de RAM, não a conseguiam aproveitar. Outra, era o facto de o DOS não conseguir fixar apenas numa partição de disco rigido, mais do que 32Mb.

Esta nova versão resolve já estes problemas e apresenta ainda alguns novos utilitários bastante úteis ao utilizador. Mas vejamos pormenorizadamente quais são os novos e melhorados aspectos deste DOS, e como tirar partido deles:

2. NOVAS CARACTERISTICAS

Em minha opinião, o melhor advento do DOS 4.0 é o quebrar da barreira dos 640Kb de RAM. É agora possivel com esta nova versão e com a norma EMS aproveitarem-se expansões de memória até 32Mb, o que é bastante, apesar de haver já equipamentos baseados no microprocessador 80386 que conseguem endereçar até 64Mb.

A EMS NO CONTEXTO DO DOS 4.0

EMS são as iniciais de Expanded Memory System e a sigla tornou-se bastante falada ultimamente desde a saída da sua última versão, por coincidência também a 4.0, que foi patrocionada por 3 grandes empresas do meio micro-informático: a Lotus Development Corporation, a Intel Corporation e a Microsoft Corporation (grupo que ficou conhecido por LIM e que verificou a grande limitação do DOS no endereçamento de apenas 640Kb de RAM , o que não lhes permitia utilizar ao máximo as capacidades dos seus produtos).

Mas afinal o que é a EMS?

Torna-se necessário, para compreendermos melhor o objectivo da EMS, retroceder um pouco ao passado dos microprocessadores compativeis. Senão vejamos...

Quando primeiramente surgiu o microprocessador Intel 8088 que só conseguia gerir 1Mb de memória, optou-se por fazer com que os primeiros 640Kb servissem de base para o funcionamento das aplicações e do sistema deixando os restantes 384Kb para servirem o funcionamento de componentes essenciais ao sistema, tal como a RAM Video. Para as aplicações correntes, isso era mais do que o necessário, mas com o passar do tempo foram surgindo aplicações maiores e novos microprocessadores, tal como o Intel 80286 e o Intel 80386, que estavam aptos a gerir bastante mais memória. Verificou-se então que, apesar do avanço na tecnologia do Hardware, o mesmo não se passou com o Software de Sistema DOS que continuava a poder gerir apenas 640Kb de RAM, porque, obrigatóriamente, tinha de se manter compativel com todas as máquinas. Foi então neste ambiente que surgiu a EMS. Como funciona? Vamos ver...

Com a intenção de contornar a barreira dos 640Kb, o sistema EMS controla uma área dentro de memoria de 1Mb que é acessivel pelo microprocessador. A memória dentro deste quadro esta ainda dividida em partes muito menores, designadas por páginas. A carta EMS é igualmente dividida em páginas correspondentes e é então possivel fazer com que estas últimas apareçam ao microprocessador em qualquer endereço. Assim, quando um programa pretende utilizar memória expandida, ele apresenta o seu pedido ao driver da EMS - o EMM (Expanded Memory Manager) que é carregado no Boot - o qual, por sua vez, selecciona as páginas disponiíveis e as reserva para uso do programa. Outros programas farão o mesmo até que toda a RAM disponível tenha sido atribuída. Quando este processo de mapeamento estiver completo, sempre que o microprocessador tiver que aceder a um determinado endereço de página, na realidade ele estará a aceder ao endereço de página correspondente na memória expandida.

Tornou-se assim possivel, com a EMS 4.0 suportar até 32Mb de RAM - permitindo o desenvolvimento de aplicações maiores (e que possam fazer uso de toda esta quantidade de memória), e ainda suportar a execução de programas em multitarefa do tipo Windows ou DesqView.

Com o DOS 4.0 deixa então de ser necessário usar drivers especiais - tanto de origem do fabricante, como desenvolvidos pelo próprio programador - passando apenas a utilizar-se os drivers para a EMS fornecidos com o próprio sistema operativo.

Através do FDISK podemos agora declarar o tamanho de uma partição em MegaBytes ou em determinada percentagem de ocupação do disco. É assim possivel, se por exemplo possuirmos um disco de 40Mb, informarmos o DOS de que queremos uma partição que ocupe 100% do espaço disponiível do disco. Teremos assim uma partição com 40Mb, o que não era possivel com as versões anteriores do MS-DOS. No caso de discos de grande capacidade (tipo 300Mb) era necessário, com as versões anteriores, criar 8 ou 9 partições de maneira a podermos aproveitar todo o disco para DOS. Com esta nova versão podemos utilizar partições até 2Gb, o que é mais do que o suficiente.

Um outro utilitário que considero bastante útil é o comando MEM. Este dá-nos um mapa de memória e é de bastante auxilio para programadores que por vezes necessitam de recorrer a outro software para obterem esta informação. Um exemplo da utilização deste comando é:

A:\>MEM /program

O resultado obtido num OLIVETTI M290 com 640Kb de base memory, 256Kb de extended memory e 128Kb de shadow memory foi o seguinte mapa:

Address	Name	Size	Type
000000		000400	Interrupt Vector
000400		000100	ROM Communication Area
000500		000200	DOS Communication Area
000700	IBMBIO	0024E0	System Program
002BE0	IBMDOS	0088B0	System Program
00B490	IBMBIO	0066E0	System Data
	DISPLAY	0036F0	DEVICE=
		0000C0	FILES=
		000100	FCBS=
		001F40	BUFFERS=
		0001C0	LASTDRIVE=
		000CD0	STACKS=
011B80	COMMAND	001640	Program
0131D0	IBMDOS	000030	— Free —
013210	COMMAND	0000A0	Environment
0132C0	MOUSE	000040	Environment
013310	APPEND	001E20	Program
015140	GRAPHICS	000050	Environment
0151A0	GRAPHICS	000950	Program
015B00	MOUSE	0023B0	Program
017EC0	MEM	000040	Environment
017F10	MEM	012F60	Program
02AE80	IBMDOS	075170	— Free —

655360 bytes total memory
655360 bytes available
557280 largest executable program size

262144 bytes total extended memory
262144 bytes available extended memory

3. INTERFACE COM O UTILIZADOR - O DOS SHELL

Para além de tudo, o novo interface do DOS com o utilizador - o DOS SHELL - é bastante agradável de se utilizar, apesar de, em minha opinião, ser um pouco lento. Mas, é claro, tudo se paga. De qualquer maneira, este novo interface põe à disposição do utilizador uma série de características que torna a vida muito mais simples para um utilizador com poucos

conhecimentos de sistema operativo. Em resumo, com o DOS 4.0 o utilizador passa a ter que saber apenas que é necessário formatar uma disquete antes de poder utilizá-la, não precisando de saber à partida qual o comando a usar e/ ou qual a sua sintaxe.

É de frizar também a facilidade com que se podem acrescentar novos programas ao sistema. O DOS SHELL permite fazê-lo com a maior das facilidades, protegê-los com uma password e ainda acrescentar-lhes um HELP com texto completamente definido pelo utilizador.

Embora a definição de uma password para a chamada de um programa impeça já, de alguma maneira, descuidos por parte de utilizadores inexperientes - tal como pedido de formatação de um disco rigido -, ela não impede, de maneira nenhuma, a entrada em sistemas protegidos com a intenção de bloquear passagens a quem não se deseja. E isto porque é sempre possivel saír do DOS SHELL para a prompt comum às outras versões do sistema operativo e daí chamar o programa. Claro que isto, para um utilizador com poucos conhecimentos pode não ser fácil, mas para um outro que esteja já habituado a trabalhar com versões anteriores, "é tão fácil como saltar à corda".*(nota do Editor: eu não sei saltar à corda)*

Para além do mais, o DOS SHELL não está à disposição de todos os utilizadores, porque só funciona em determinadas cartas compativeis com EGA e VGA. (Tenho a vaga sensação de que a MicroSoft está a tentar proteger-se um pouco contra as cópias piratas, porque, para além deste problema, o DOS 4.0 trás ainda um número de série que é sempre visivel quando se faz um DIR).

Outro ponto significativo na utilização do DOS SHELL é a possibilidade de podermos visualizar simultaneamente dois directórios (o que, por vezes, é de grande ajuda), e de podermos seleccionar ficheiros para que, sobre eles possamos executar operações básicas tal como cópias.

É ainda possivel escolher como configuração de cores a utilizar, uma das quatro opções que este interface nos põe à disposição. Os quatro conjuntos de cores baseiam-se em tons de branco, cinzento ou azul, cores que, à partida ferem menos a vista. Não é no entanto possivel definir outras configurações, tal como acontece no Windows.

4. MELHORAMENTOS

Os grandes melhoramentos (para além dos já mencionados) nesta versão do DOS referem-se essencialmente às mensagens para o utilizador, e à prevenção de anulação de ficheiros com o comando DEL/ERASE. É agora possivel - através de um parâmetro passado ao comando - perguntar se quer apagar ou não antes da anulação ser definitiva.

A:\>DEL XPTO.TXT /p

A:\XPTO.TXT, Delete (Y/N)?

O comando TREE apresenta agora uma estrutura em arvore - tipo PCTOOLS - bastante mais legível do que as versões antigas e o comando BACKUP poderá eventualmente formatar disquetes que não tenham sido formatadas antes.

É tambem visivel um novo aumento de drivers nesta versão do DOS, e alguns comandos, tais como:

GENERALIDADES SOBRE DOS SHELL

O Batch file para utilização do DOS SHELL completo é o seguinte:

```
@SHELLB DOSSHELL
@IF ERRORLEVEL 255 GOTO END
@BREAK=OFF
@SHELLC/MOS:PCIBMDRV.MOS/TRAN/COLOR/DOS/MENU/
MUL/SND/MEU:SHELL.MEU/CLR:SHELL.CLR/PROMPT/MAINT/
EXIT/SWAP/DATE
:END
@BREAK=ON
```

Desta maneira é então possivel ligar o computador e automaticamente carregar o DOSSHELL, desde que este seja chamado através do AUTOEXEC.BAT. Depois disto é bastante simples utilizar este interface grafico. Teclas a conhecer são a tecla F10, o TAB e, claro, a tecla de ENTER ou CR. Com estes conhecimentos basicos esta-se apto a passear pelos menus sem grandes dificuldades.

Existem, à partida dois tipos de menus:

- Um menu de linha, em cima, o qual pode ser acedido pela tecla F10 e que pode executar funções como mandar executar um determinado programa, adicionar um novo programa ao disco, etc (opção Program); o mesmo pode acontecer se escolhermos a opção Group ou ainda seleccionarmos a opção de abandono do DOS SHELL.
- Um menu em coluna, onde se fica colocado assim que se entra no DOS SHELL, e que permite ir até ao prompt normal do sistema (Command Prompt), visualizar o conteúdo de drives ou subdirectorias (File System), executar alguns utilitários do DOS (DOS Utilities) e ainda uma opção de mudança de cores (Change Colors). Podemos seleccionar uma destas opções com a maior das simplicidades usando as teclas de movimento do cursor. É então fácil acrescentar a este menu novas opções que poderão chamar novos programas que deseja correr neste ambiente.

O ambiente escolhido através da opção File System é de bastante utilidade pois põe à disposição do utilizador utilitários gerais de tratamento de ficheiros - (Opção File, onde figuram comandos tais como o Print, Rename, etc...); opções de ordenação de ficheiros, opções de configuração - tais como confirmação de anular de ficheiros, confirmação de substituição de ficheiros e seleccionamento através de directorias -, informações acerca de ficheiros - nome e atributos -, drive e directoria seleccionada com nome, tamanho e nº de ficheiros là existentes e ainda toda a informação acerca do disco - (Opção Options);

Com a opção Arrange podemos optar pela visualização de uma simples drive ou directoria ou de duas simultaneamente, ou ainda de todas as informações mencionadas na última parte do parágrafo anterior e de todos os ficheiros.

Finalmente, usando a opção de DOS Utilities é bastante fácil formatar, copiar disquetes ou fazer backups.

- o GRAPHICS, foi melhorado para as placas EGA e VGA , sendo agora possivel imprimir écrans assim como produzir outputs em 19 tons de cinzento ou 8 cores;

- o CHKDSK que anota tamanho e número de clusters disponíveis;

- e ainda o parâmetro BUFFERS do CONFIG.SYS foi também aumentado para um limite maximo de 10000 buffers em memória expandida.

5. OS MAUS BOCADOS DO DOS 4.0

Quando me deram a disquete com o sistema para fazer a análise, pensei que apenas fosse necessário ligar o computador com a disquete na drive, tal qual estamos habituados, para que me surgisse o agradável interface do DOS SHELL. Enganei-me redondamente! Suei um bom bocado para lá conseguir chegar.

Mas vejamos, para se conseguir lá chegar é necessário chamar o comando SHELLC com um determinado número de parâmetros como se pode ver no quadro de "Generalidades sobre DOS SHELL". A maneira mais simples, claro, é colocar os comandos dentro de um ficheiro batch e chamar por ele sempre que o pretendermos executar. De qualquer maneira, penso que não é nada facil de instalar embora as suas vantagens sejam por demais evidentes.

Quanto ao RATO bem tentei, mas não consegui instalá-lo no curto espaço de tempo que tive para fazer esta análise.

Experimentei também executar o DOS SHELL em computadores com placa CGA e EGA, mas a resposta foi bastante clara:

EGA ou VGA adaptor with 256Kb memory required to display graphics

Isto pode ser uma limitação, mas a tendência no mercado é a evolução para placas gráficas de alta resolução.

Nota-se contudo uma grande lentidão no carregamento do DOS no BOOT embora a MicroSoft proclame que nesta versão a existência de uma maior área de FAT (File Allocation Table) acelera o acesso aos ficheiros.

6. CONCLUSÕES

Esta nova versão do DOS é, de modo geral, mesmo uma nova versão, pelo menos no que toca ao interface com o utilizador. Os problemas que resolveu em termos de memória RAM, disco rigído e, quem sabe?, multitarefa, veio fazer do DOS uma alternativa viavel ao OS/2, que em termos de limitações se assemelha muito à antiga versão do DOS: o DOS 3.3. Uma coisa é certa, programas utilitários para o DOS tipo Norton Commander vão começar a sofrer as consequências deste novo DOS.

Depois disto, só resta acrescentar que só falta mesmo o DOS multiposto! Para quando o DOS 5.0?

Paulo Pinheiro

AUMENTE A SUA LIVRARIA DE C

(III e última Parte)

NUMA era informática, em que a segurança de dados assume um papel fundamental, torna-se praticamente obrigatória a presença dos próximos dois programas. Trata-se pois de um codificador e de um descodificador de ficheiros de texto (ASCII).

A função **code** abre um ficheiro de texto, previamente direccinado pelo utilizador, lê caracter a caracter, e posiciona o complemento (~) desse mesmo caracter num 2º ficheiro, este aberto como binário, já que o operador complemento dá origem a caracteres não alfanuméricos.

Com a função **dcode** acontece o oposto, sendo aberto um ficheiro binário, donde é lido um byte que é posteriormente complementado e introduzido num ficheiro de texto, já que se tratam de bytes previamente codificados através da função **code** que só não dá origem a caracteres alfanuméricos.

```
CODE.C

#include <stdio.h>

main (argc,argv)

 int argc;
 char *argv [];

{

 FILE *in,*out;
 char ch,ch1;

   if (argc!=3)  {
       puts ("Parametros inválidos  ...");
       exit (0);
   }

   if (( in = fopen ( argv[1],"r")) == NULL )   {
         printf ( "Erro na abertura do ficheiro %s ...",argv[1] );
         exit (0);
   }

   if (( out = fopen ( argv[2],"wb")) == NULL )  {
         printf ( "Erro na abertura do ficheiro %s ...",argv[2] );
         exit (0);
   }

   ch = getc (in);

   while (!feof (in)) {
                       ch1 = ~ch;
                       putc (ch1,out);
                       ch = getc (in);
   }

   puts ("Ficheiro codificado");

   fclose (in);
   fclose (out);


      /* A função code codifica qualquer ficheiro de texto utilizando o
         operador ~ para fazer o complemento do caracter a ser modificado.
             O operador ~ funciona como que o inverso => todos os bits
         que estão a 1 passam a 0, e todos os que estão a 0 passa para 1.
         O operador ~ dá origem a caracters não alfanuméricos, daí a razão
         de um ficheiro binário para o 'output' */

DCODE.C


#include <stdio.h>

main (argc,argv)

 int argc;
 char *argv [];

{

 FILE *in,*out;
 char ch,ch1;

   if (argc!=3)  {
       puts ("Parametros inválidos ...");
       exit (0);
   }

   if (( in = fopen ( argv[1],"rb")) == NULL )   {
         printf ( "Erro na abertura do ficheiro %s ...",argv[1] );
         exit (0);
   }

   if (( out = fopen ( argv[2],"w")) == NULL )  {
         printf ( "Erro na abertura do ficheiro %s ...",argv[2] );
         exit (0);
   }
```

```
        ch = getc (in);

        while (!feof (in)) {
                        ch1=~ch;
                        putc ( ch1,out);
                        ch=getc (in);
        }

        puts ("Ficheiro descodificado");

      fclose (in);

      fclose (out);

}

        /* A função dcode é análoga à função code, descodificando qualquer
        ficheiro de texto previamente codificado utilizando o operador ~ para
        fazer o complemento do caracter a ser modificado.
              O operador ~ funciona como que o inverso => todos os bits
          que estão a 1 passam a 0, e todos os que estão a 0 passa para 1.
          O operador ~ só dá origem a caracters alfanuméricos, visto que o
          ficheiro anteriormente codificado era um ficheiro de texto, daí
          a razão de um ficheiro de texto para o 'output' */
```

Poderoso como só o C pode ser, é possível os programadores criarem "macros" e constantes dentro de programas através da directiva **# define**.

A diferença entre um macro e uma constante, é que ao passo que a constante toma sempre o mesmo valor ao longo de todo o programa, o macro toma valores diferentes, consoante a forma como for utilizado. Exemplo: a constante ON é traduzida pela expressão **# define On (valor)** forçando assim o compilador a uma imediata substituição da palavra **On,** pelo valor **(valor)**, sempre que esta for encontrada dentro do programa. No macro **# define quadrado (x) (x)*(x)**, o compilador vai substituir quadrado (x) por (x)*(x), assumindo assim diversos valores, consoante a variável x.

Os macros e as constantes são pois muito importantes, não só por aumentarem a portabilidade dos programas, como também por darem um aspecto mais estruturado ao programa.

É com este fim que aqui expomos algumas constantes e alguns macros, que nos parecem do interesse geral e de grande uso para qualquer programador de C.

```
#define NULL '\0'                        /* define o caracter NULL */

#define NEWLINE '\n'                     /* define carriage return/line
                                         feed */

#define BACKSPACE '\b'
#define CARRIAGE_RETURN '\r'
#define FORMFEED '\f'
#define TRUE 1                           /* Boolean TRUE em C */
#define FALSE 0                          /* Boolean FALSE */
#define PI 3.141592653
#define MAX_STRING 255                   /* dimenção máxima de
                                         uma string */

#define NO_ERROR 0

define EOL '\n'                          /* define o  fim da linha */
#define EOF -1                           /* define o fim do ficheiro */
#define ERRO -1                          /* define o valor de retorno
                                         de ERRO */

#define BLANK ' '
#define print_character(x) putch(x)
#define get_character(x) getch(x)

#define IO_ERROR -1                      /* valor retornado se o
                                         utilizador introduz dados
                                         errados através de teclado */


#define BLACK 0
#define BLUE 1
#define GREEN 2
#define CYAN 3
#define RED 4
#define MAGENTA 5
#define BROWN 6
#define LIGHTGRAY 7
#define DARKGRAY 8
#define LIGHTBLUE 9
#define LIGHTGREEN 10
#define LIGHTCYAN 11
#define LIGHTRED 12
#define LIGHTMAGENTA 13
#define YELLOW 14
#define WHITE 15

/*
 * NOME: percentagem(x,y)
 *
 * FUNÇAO: Retorna a percentagem de x / y.
 *
 * EXEMPLOS: a = percentagem(3.0, 4.0);   atribui a a o valor 75.0
 *
 * VARIAVEIS UTILIZADAS: x: o dividendo.
 *                          y: o divisor.
 *
 * PSEUDO CóDIGO: multiplica x por 100.0 (também convertee para float)
 *                divide x por y.
 *
 */

#define percentagem(x, y) (100.0 * x / y)

/*-----------------------------------------------------------------------*/

/*
 * NOME: abs_val(x)
 *
 * FUNÇAO: Retorna o valor absoluto do valor contido
 *         na variável x.
 *
 * EXEMPLOS: x = abs_val(-2);        atribui a x o valor 2
 *           x = abs_val(4);         atribui a x o valor 4
 *           x = abs_val(4 - 13);    atribui a x o valor 9
 *
 * VARIAVEIS UTILIZADAS: x: a expressão para a qual se retorna
 *                          o valor absoluto.
 *
 * PSEUDO CóDIGO:  se (x é positivo)
 *                   retorna (x)
 *                 senão
 *                   retorna (-x)
 *
 */

#define abs_val(x) ((x >= 0) ? x : 0 - x)

/*-----------------------------------------------------------------------*/

/*
 * NOME: max(x, y)
 *
 * FUNÇAO: Retorna o valor máximo contido em x or y.
 *
 * EXEMPLOS:    x = max(3, 4);          atribui a x o valor 4
 *              x = max(4, 4);          atribui a x o valor 4
 *              x = max(4*3, 4+3);      atribui a x o valor 12
 *
 * VARIAVEIS UTILIZADAS: x, y: valores a comparar para o máximo.
 *
 * PSEUDO CóDIGO:  se (x é maior que y)
 *                      retorna (x)
```

```
 *              senão
 *                   retorna (y)
 *
 */

#define max(x, y) ((x > y) ? x : y)

/*
 * NOME: min(x, y)
 *
 * FUNÇAO: Retorna o valor minimo contido em x ou y.
 *
 * EXEMPLOS:   x = min(3, 4);      atribui a x o valor 3
 *             x = min(3*3, 3+3);  atribui a x o valor 6
 *
 * VARIAVEIS UTILIZADAS: x, y: valores a comparar para minimo.
 *
 * PSEUDO CóDIGO:   se (x é menor que y)
 *                     retorna (x)
 *                  senão
 *                     retorna (y)
 *
 */

#define min(x, y) ((x < y) ? x: y)

/*--------------------------------------------------------------------------*/

/*
 * NOME: e_par(x)
 *
 * FUNÇAO: Retorna 1 se o valor contido em x for par, senão
 *         retorna o valor 0. O 'mod operator' (%) retorna o
 *         resto da divisão inteira de x e 2. Se o resto
 *         for 0, então x é impar. Se o resto é 1 então
 *         x é par.
 *
 * EXEMPLOS:    se (e_par(3))      retorna 1
 *              se (e_par(444))    retorna 0
 *
 * VARIAVEIS UTILIZADAS: x: variável examinada para ver se é par.
 *
 * PSEUDO CóDIGO:  se (x é par)
 *                   retorna (1)      -- TRUE booleano
 *                 senão
 *                   retorna (0)      -- FALSE Booleano
 *
 */

#define e_par(x) ((x % 2 == 1) ? 1 : 0)

/*--------------------------------------------------------------------------*/

/*
 * NOME: e_impar(x)
 *
 * FUNÇAO: Retorna 1 se o valor contido em x é impar, senão
 *         retorna o valor 0. O 'mod operator' (%) retorna
 *         o resto da divisão inteira de x e 2. Se o resto
 *         é 0, então x é impar. Se o resto é 1
 *         então x é par.
 *
 * EXEMPLOS:    se (e_impar(30))     retorna 1
 *              se (e_impar(4445))   retorna 0
 *
 * VARIAVEIS UTILIZADAS: x: variável examinada para ver se é impar.
 *
 * PSEUDO CóDIGO:  se (x é impar)
 *                   retorna (1)      -- TRUE booleano
 *                 senão
 *                   retorna (0)      -- FALSE Booleano
 *
 */

#define e_impar(x) ((x % 2 == 0) ? 1 : 0)

/*--------------------------------------------------------------------------*/

/*
 * NOME: round_off(x)
 *
 * FUNÇAO: Arredonda o valor contido em x.
 *
 * EXEMPLOS:   i = round_off(3.443);      atribui a i o valor 3
 *             i = round_off(3.55);       atribui a i o valor 4
 *
 * VARIAVEIS UTILIZADAS:  x: contem o valor a arredondar.
 *
 * PSEUDO CóDIGO:  Soma 0.5 ao valor a arredondar
 *                 Fazer um 'cast' de tipo int ao valor
 *
 */

#define round_off(x) ((int)(x + 0.5))

/*--------------------------------------------------------------------------*/

/*
 * NOME: cortar(x)
 *
 * FUNÇAO: Corta a parte decimal do valor contido em x.
 *
 * EXEMPLOS:   i = cortar(3.443);          atribui a i o valor 3
 *             i = cortar(0.11 - 0.05);    atribui a i o valor 0
 *
 * VARIAVEIS UTILIZADAS:  x: contem o valor a cortar.
 *
 * PSEUDO CóDIGO:  Fazer um 'cast' de tipo int ao valor
 *
 */

#define cortar(x)  ((int)(x))

/*--------------------------------------------------------------------------*/

/*
 * NOME:  resto(x, y)
 *
 * FUNÇAO:  Usa o 'mod operator' para retornar o resto
 *          da divisão inteira dos valoers contidos em x e y.
 *
 * VARIAVEIS UTILIZADAS: x: o dividend.
 *                       y: o divisor.
 *
 * PSEUDO CóDIGO:  mod o valor contido em x por o valor contido
 *                 em y para produzir o resto
 *
 */

#define resto(x, y) (x % y)

/*--------------------------------------------------------------------------*/

/*
 * NOME: quadrado(x)
 *
 * FUNÇAO: Retorna o quadrado do valor contido em x.
 *
 * EXAMPLE: y = quadrado(9);      atribui a y o valor 81
 *
 * VARIAVEIS UTILIZADAS: x: contem o valor a ser elevado quadrado.
 *
 * PSEUDO CóDIGO: multiplica o valor contido em x por si mesmo.
 *
 */

#define quadrado (x) ((x)*(x))

/*--------------------------------------------------------------------------*/

/*
 * NOME: cubo(x)
 *
 * FUNÇAO: Retorna o cubo do valor contido em x.
 *
 * EXAMPLE: y = cubo(9); atribui a y o valor 729 (9 * 9 * 9)
 *
 * VARIAVEIS UTILIZADAS: x: contem o valor a ser elevado ao cubo.
 *
 * PSEUDO CóDIGO: multiplica o valor contido em x três vezes por si mesmo.
 *                (cubo(4) ==> 4 * 4 * 4)
 */

#define cubo(x) ((x)*(x)*(x))

/*--------------------------------------------------------------------------*/

/*
 * NOME: XOR(x,y)
 *
 * FUNÇAO: Retorna o or exclusivo dos bits contidos
 *         nas variáveis x e y.
 *
 *
 *          O or exclusivo utiliza a seguinte tabela de verdade
 *
```

```
*               x   y   resultado
*               0   0     0
*               0   1     1
*               1   0     1
*               1   1     0
*
* VARIAVEIS UTILIZADAS: x, y: contêm os valores a sofrer o or exclusivo.
*
* PSEUDO CóDIGO: inverte (ones complement(~)) o valor em y
*                AND x e o valor de y invertido
*
*               inverte o valor contido em x
*               AND o valor em y com o valor invertido de x
*
*               OR o resultado das expressões acima referidas
*
*/

#define XOR(x, y) ((x & ~y) | (~x & y))
/*--------------------------------------------------------------------------*/

/*
* NOME: NOT(x)
*
* FUNÇÃO: Retorna o Boolean NOT do valor contido em x.
*         Se x é TRUE, FALSE é retornado.  Se x é FALSE, TRUE
*           é retornado.
*
* EXAMPLE: enquanto (NOT(done))
*
* VARIAVEIS UTILIZADAS: x: contem o valor to NOT.
*
* PSEUDO CóDIGO: se (x é > 0) (Boolean false)
*                   retorna (0)                  -- FALSE Booleano
*                senão
*                   retorna (1)                  -- TRUE booleano
*
*/

#define NOT(x) ((x != 0) ? 0 : 1)

/*--------------------------------------------------------------------------*/

/*
* NOME: bit_mask(x, y)
*
* FUNÇÃO: Executa um AND bit a bit dos valores contidos em x e y
*
* EXAMPLE: mask = bit_mask(1, 3); atribui a mask o valor 1
*
*            0001 & 0011 produz 0001
*
* VARIAVEIS UTILIZADAS: x, y: contêm valores a fazer o bit_mask.
*
* PSEUDO CóDIGO: Executa um AND (&) bit a bit a cada bit
*                contido nos valores de x e y.
*
*/

#define bit_mask(x, y) (x & y)

/*--------------------------------------------------------------------------*/

/*
* NOME: e_digito(x)
*
* FUNÇÃO: Retorna 1 se x é a caracter entre '0' to '9'
*         caso contrário retorna 0.
*
* EXEMPLOS:   se (e_digito('7'))      retorna 1
*             se (e_digito('a'))      retorna 0
*
* VARIAVEIS UTILIZADAS: x: variável examinada para ver se é digito.
*
* PSEUDO CóDIGO:  se (x está entre '0' to '9')
*                    retorna (1)   -- TRUE booleano
*                 senão
*                    retorna (0)   -- FALSE Booleano
*
*/

#define e_digito(x)  ((x >= '0' && x <= '9') ? 1 : 0)
/*--------------------------------------------------------------------------*/

/*
* NOME: e_maiuscula(x)
*
* FUNÇÃO: Retorna 1 se x é uma letra maiúscula,
*         caso contrário retorna o valor 0.
*
* EXEMPLOS:   se (e_maiuscula('S'))     retorna 1
*             se (e_maiuscula('9'))     retorna 0
*             se (e_maiuscula('a'))     retorna 0
*
* VARIAVEIS UTILIZADAS: x: variável examinada para ver se é maiúscula.
*
```

```
*  PSEUDO CÓDIGO:  se (x está entre 'A' to 'Z')
*                     retorna (1)          -- TRUE booleano
*                  senão
*                     retorna (0)          -- FALSE Booleano
*
*/

#define e_maiuscula(x) ((x >= 'A' && x <= 'Z') ? 1 : 0)

/*---------------------------------------------------------------------------*

/*
*  NOME: e_minuscula(x)
*
*  FUNÇÃO: Retorna 1 se x é uma letra minuscula letter,
*          caso contrário retorna o valor 0.
*
*  EXEMPLOS:   se (e_minuscula('s'))     retorna 1
*              se (e_minuscula('9'))     retorna 0
*              se (e_minuscula('A'))     retorna 0
*
*  VARIAVEIS UTILIZADAS: x: variável examinada para ver se é minuscula.
*
*  PSEUDO CÓDIGO:  se (x é está entre 'a' 'to 'z')
*                     retorna (1)   --  TRUE booleano
*                  senão
*                     retorna (0)   --  FALSE Booleano
*
*/

#define e_minuscula(x) ((x >= 'a' && x <= 'z') ? 1 : 0)

/*---------------------------------------------------------------------------*/

/*
*  NOME: e_espaço(x)
*
*  FUNÇÃO: Retorna 1 se o valor contido em x é um espaço,
*          caso contrário retorna 0.
*
*  EXEMPLOS:  se (e_espaço(' '))      retorna 1
*             se (e_espaço('a'))      retorna 0
*
*             enquanto (e_espaço(string[index++]))  -- salta os espaços
*
*  VARIAVEIS UTILIZADAS: x: variável a ser examinada para ver se é espaço.
*
*  PSEUDO CÓDIGO:  se (x é espaço)
*                     retorna (1) --  TRUE booleano
*                  senão
*                     retorna (0) --  FALSE Booleano
*
*/

#define e_espaço(x) ((x == ' ') ? 1 : 0)

/*---------------------------------------------------------------------------*/

/*
*  NOME: BELL
*
*  FUNÇÃO:  Toca a campainha do computador.
*
*  EXEMPLOS:  for (i = 1; i < 10; i++)
*                 BELL;
*
*  VARIAVEIS UTILIZADAS: Nenhuma.
*
*  PSEUDO CÓDIGO: Utiliza a função putchar para imprimir um caracter ASCII 7
*
*/

#define BELL putchar(7)

/*---------------------------------------------------------------------------*/

/*
*  NOME: para_maiuscula(x)
*
*  FUNÇÃO: Converte o caracter contido na variável x
*          para maiúscula se é uma letra minuscula,
*          caso contrário o caracter não é modificado.
*
```

```
 *
 * PSEUDO CÓDIGO: se (x é > 32 e x é < 127)
 *                   retorna (1)  -- TRUE booleano
 *               senão
 *                   retorna (0)  -- FALSE Booleano
 *
 */

#define e_ascii_imprimivel(x)  ((x > 32 && x < 127) ? 1 : 0)

/*----------------------------------------------------------------------------*/

/*
 * NOME: para_ascii(x)
 *
 * FUNÇAO: Converte um inteiro entre 0 - 9 para a sua
 *         representação em caracter ASCII.  Retorna
 *         o valor -1 se o valor não está entre 0 - 9
 *
 * EXEMPLO: letra = para_ascii(7): atribui a letra o valor 55
 *
 * VARIAVEIS UTILIZADAS: x: contem o valor inteiro a converter.
 *
 * PSEUDO CÓDIGO: se (x é >= 0 e x é <= 9)
 *                  retorna (o a representação ASCII somando '0'
 *               senão
 *                  retorna (-1)
 *
 */

#define para_ascii(x) ((x >= 0 && x <= 9) ? x + '0' : -1)

/*----------------------------------------------------------------------------*/

/*
 * NOME: e_binário(x)
 *
 * FUNÇAO: Retorna TRUE se x contem um '1' or '0',
 *         caso contrário retorna FALSE.
 *
 * EXEMPLO: se (e_binário('4'))    retorna 0
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter ASCII a examinar examine.
 *
 * PSEUDO CÓDIGO: se (x é igual a '0' ou x é igual a '1')
 *                  retorna (1)  -- TRUE booleano
 *               senão
 *                  retorna (0)  -- FALSE Booleano
 *
 */

#define e_binário(x) ((x == '0' || x == '1') ? 1 : 0)

/*----------------------------------------------------------------------------*/

/*
 * NOME: e_hexadecimal(x)
 *
 * FUNÇAO: Retorna 1 se o valor contido em x é hexadecimal
 *         (entre 0 - 9 ou A - F), caso contrário retorna 0.
 *
 * EXEMPLO: enquanto (e_hexadecimal(x))
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter ASCII a examinar.
 *
 * PSEUDO CÓDIGO: se (x é um digito ou (x >= 'A' e x <= 'F'))
 *                  retorna (1)  -- TRUE booleano
 *               senão
 *                  retorna (0)  -- FALSE Booleano
 *
 */

#define e_hexadecimal(x) ((e_digito(x) || (x >= 'A' && x <= 'F')) ? 1 : 0)

/*----------------------------------------------------------------------------*/

/*
 * NOME: e_octal(x)
 *
 * FUNÇAO: Retorna 1 se o valor contido em x é octal
 *         (entre '0' - '7'), caso contrário retorna 0.
 *
 * EXEMPLO: enquanto (e_octal(x))
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter ASCII a ser examinado.
 *
 * PSEUDO CÓDIGO: se (x é >= '0' e x <= '7')
 *                  retorna (1)  -- TRUE booleano
 *               senão
 *                  retorna (0)  -- FALSE Booleano
 *
 */

#define e_octal(x) ((x >= '0' && x <= '7') ? 1 : 0)
```

```
 * EXEMPLOS: para_maiuscula('4');  retorna '4'
 *           para_maiuscula('a');  retorna 'A'
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter a converter para maiúscula.
 *
 *
 * PSEUDO CÓDIGO:  se (x é minuscula)
 *                   converte para maiuscula
 *                 senão
 *                   retorna letra
 *
 */

#define para_maiuscula(x)  ((e_minuscula(x)) ? x - 'a' + 'A' : x)

/*----------------------------------------------------------------------------*/

/*
 * NOME: para_minuscula(x)
 *
 * FUNÇAO: Converte o caracter contido na variável x
 *         para minuscula se é uma letra maiuscula,
 *         caso contrário o caracter não é modificado.
 *
 * EXEMPLOS: para_minuscula('4');  retorna '4'
 *           para_minuscula('A');  retorna 'a'
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter a converter para minuscula
 *
 *
 * PSEUDO CÓDIGO:  se(x é maiúscula)
 *                   converte para minuscula
 *                 senão
 *                   retorna letra
 *
 */

#define to_minuscula(x) ((e_maiuscula(x)) ? x + 'a' - 'A' : x)

/*----------------------------------------------------------------------------*/

/*
 * NOME: para_decimal(x)
 *
 * FUNÇAO: Converte o digito ASCII para o correspondente valor
 *         decimal.  Assume-se que x já foi verificado como
 *         sendo um digito pelo macro e_digito().
 *
 * EXEMPLO: x = para_decimal('3'); atribui a x o valor 3
 *
 * VARIAVEIS UTILIZADAS: x: caracter a converter para decimal.
 *
 * PSEUDO CÓDIGO: converte x para decimal subtraindo '0'
 *
 *                 '3' - '0' = 51 - 48 (Decimal) = 3
 *
 */

#define para_decimal(x) (x - '0')

/*----------------------------------------------------------------------------*/

/*
 * NOME: e_control(x)
 *
 * FUNÇAO: Retorna 1 se o valor contido em x é um caracter de control
 *         de outra forma retorna 0.
 *
 * EXEMPLO: e_control('a')  retorna 0
 *          e_control(14)   retorna 1
 *
 * VARIAVEIS UTILIZADAS: x: letra a ser examinada.
 *
 * PSEUDO CÓDIGO: se (x é > 0 e x é < 32) -- caracteres de control ASCII
 *                  retorna (1)                   -- TRUE booleano
 *               senão
 *                  retorna (0)                   -- FALSE Booleano
 *
 */

#define e_control(x) ((x > 0 && x < 32) ? 1 : 0)

/*----------------------------------------------------------------------------*/

/*
 * NOME: e_ascii_imprimivel(x)
 *
 * FUNÇAO: Retorna 1 se x é a e um caracter ASCII imprimivel,
 *         ou retorna 0 se o caracter não é imprimivel.
 *
 * EXEMPLO: se (e_ascii_imprimivel(44))
 *
 * VARIAVEIS UTILIZADAS: x: contem o caracter ASCII a ser examinado.
```

O MS-DOS POR DENTRO

Este é o primeiro duma série de artigos sobre a maneira de usar as rotinas do MSDOS, sendo este um tema que é absolutamente necessário conhecer a fundo para a execução dum programa em Assembler ou qualquer outra linguagem explorando todas as características deste sistema operativo.

MAIS de metade dos leitores da AM perguntam a si próprios qual a utilidade de tal facto, ou, pior ainda, não fazem a mínima ideia ao que nos referimos quando falamos em "usar as rotinas do MS-DOS".

É especialmente para esses utilizadores que dirigimos o início deste primeiro artigo tentando "aliciá-los" para espreitarem um pouco para o mundo dos programadores profissionais.

Começo por referir que sempre que possível e que tal for pertinente vão apresentar-se pequenos exemplos de rotinas escritas em C e uma ou outra em Assembler, note-se no entanto que o leitor não necessita obrigatoriamente de conhecer alguma destas linguagens para perceber estas rotinas embora uns certos conhecimentos de C fossem úteis. Espero que no fim desta série de artigos todos os leitores se sintam "aliciádos" para deixar por momentos o BASIC e tentem aprender a C, dado que em BASIC não se pode aceder a certas facilidades sendo aliás este o factor pelo qual escolhi o C além de que é bastante mais acessível, acrescente-se.

O QUE É UM SISTEMA OPERATIVO?

Não, os sistemas operativos não servem só para correr programas, eles executam muitas mais tarefas que o utilizador comum nem se quer desconfia que existem.

A função dum sistema operativo genericamente é a de fornecer as ferramentas para o trabalho de todos os utilziadores dum sistema. Assim, fornece todas as "comodidades" que geralmente damos por garantidas, como por exemplo uma tarefa aparentemente tão simples como escrever um caracter no ecrã.

Agarrando neste último exemplo, muita gente não se apercebe que quando faz um simples comando PRINT "OLA" no BASIC, o compilador ou interpretador da linguagem chama uma rotina do DOS para escrever uma cadeia de caracteres. O DOS por sua vez vai escrever cada caracter utilizando a ROM do computador, que por sua vez vai endereçar a RAM de vídeo, que por último vai ser lida pela placa gráfica que envia essa informação para o ecrã (ufff!). E, no entanto, todo este trabalho decorre sem que o utilizador dê conta dele, afinal ele apenas fez um PRINT!

Torna-se pertinente neste ponto distinguir claramente quais as tarefas especificamente executadas pelo S. O. bem como por outro tipo de Software, para tal, vou começar por explicar qual a definição dum computador dum ponto de vista de funcionamento e quais as partes que o constituem.

Para melhor entender o que faz o MS-DOS vamos usar um simples esquema bastante elucidativo, que geralmente é usado em literatura da especialidade.

Como se pode ser pela fig.1, o computador é visto como uma cebola que é constituída por várias camadas das quais o utilizador só "vê" a última camada, ignorando o que se passa nas camadas interiores.

As camadas constituintes da "cebola" são as seguintes:

1. Camada física (Hardware) — este é o centro da "cebola" e nele se englobam genericamente todos os recursos do computador incluindo memórias, discos, teclado, etc.

2. Basic Input / Output System (BIOS) para aceder a todos os recursos do Hardware directamente da melhor maneira é necessário ter um software que conheça bem o "dito cujo". Este software é constituído em parte pela ROM que vem com o computador, a que geralmente se dá o nome de ROM BIOS. A ROM BIOS encarrega-se de fazer o interface entre as camadas mais altas (mais exteriores no modelo da cebola) e o Hardware. O trabalho da BIOS é fazer tarefas que nos podem parecer esotéricas, tais como ler um caracter do teclado ou enviar um caracter para a impressora.

A consequência prática do facto de existir esta informação na ROM é que esta é a única camada que precisa

obrigatoriamente de conhecer o Hardware aliviando todas as outras camadas deste "peso". De facto, quem já tentou programar um Controlador de Disco, por exemplo, sabe ao que me refiro quando digo que esta é uma tarefa "pesada" e para adultos de sólida formação moral!

Particularizando para o caso dos IBM PC's, esta camada está implementada na ROM como já referi sendo esse facto previsível pois cada fabricante de compatíveis pode produzir alterações no Hardware dos seus computadores, bastando escrever uma ROM BIOS tendo em conta essas alterações para o computador se tornar 100% compatível. Refira-se que foi este um dos factos que mais contribuiram para o sucesso dos PC's dado que através deste processo pode fugir-se ao pagamento de muitos "royalties" à IBM e outras companhias não deixando de produzir Hardware quase igual ao do IBM PC original

3. Sistema Operativo — é aqui que finalmente chegamos ao nosso velho amigo MS-DOS.

Chegamos enfim à difícil e ingrata tarefa de explicar o que realmente faz um sistema operativo (não, não, não serve só para correr o BASIC ou gerir ficheiros!).

As principais tarefas desempenhadas por um sistema operativo são:

— Efectuar uma gestão da memória disponível no computador, evitando assim "colisões" entre programas por usarem os mesmos endereços em memória, bem como evitando que se exceda a capacidade total da memória devolvendo um erro quando isso acontecer. Tem também que ter em conta que quando um programa acaba, se deve libertar toda a memória por ele ocupada afim de poder ser usada por outro programa.

— Trata de todos os assuntos relacionados com a gestão de ficheiros "assentando" num bloco de notas, a File Allocation Table (FAT), a localização de cada ficheiro bem como todos os seus atríbutos inerentes. Imagine-se o que seria se o utilizador cada vez que quizesse ir buscar um ficheiro a uma disquete tivesse que anotar todos os sectores em que esse ficheiro estava localizado!

Deste modo apenas o sistema operativo tem a noção de como se representa um ficheiro e como ir buscá-lo ao dispositivo de armazenamento. Note-se que para ler cada ficheiro o S.O. vai chamar a BIOS as vezes necessárias para ler cada sector da disquete ou disco.

Como curiosidade, note-se que nos dispositivos de memória secundária os ficheiros estão divididos, pelo que o que designa-se por "cluster" e que é definido como o espaço mais pequeno que cada ficheiro pode ocupar na memória secundária mesmo que esse ficheiro tenha só um byte!

— Faz um tratamento de Input/Output chamando as rotinas da BIOS, isto é, aproveita os conhecimentos que o utilizador ou as aplicações possam fazer um aproveitamento maior de todos esses recursos. Esta afirmação vai tornar-se mais nítida para muitos leitores desconfiados após apresentarmos alguns exemplos bem simples e elucidativos...

— Finalmente, faz o que toda a gente já conhece que é executar as aplicações que são desenvolvidas para o computador, sendo no caso do DOS todos os ficheiros com extensão .COM, .EXE ou .BAT como o leitor já sabe.

Vamos pois assim tratar apenas do MS-DOS ignorando por as outras camadas.

Refira-se que em geral costuma-se

dividir ainda um S.O. em termos funcionais em duas sub-camadas:

— Extended Input/Output System (XIOS) — nesta sub-camada estão localizadas as rotinas de manipulação de ficheiros e de dispositivos. Uma tendência actual é a de se considerar todos os periféricos como ficheiros permitindo uma transferência de dados da mesma maneira do que para ficheiros normais como se verifica no caso do sistema operativo UNIX. Esta uniformização na manipulação dos vários periféricos contribui muito decisivamente para a facilidade da sua utilização. No caso do MSDOS esta camada encontra-se no ficheiro IBMBIOS.SYS e em parte do ficheiro IBMDOS.SYS.

— Human Interface (HINT) — constitui a camada de interface entre o S.O. e o utilizador. Está aqui implementada o interpretador de comandos bem como o modo de correr aplicações. No MSDOS esta camada encontra-se na totalidade do ficheiro COMAND.COM bem como parte do IBMDOS.SYS.

COMO É QUE AS CAMADAS COMUNICAM ENTRE SI?

A divisão por camadas atrás referida não servíria de nada, se não existissem mecanismos eficientes para cada camada comunicar entre si. Note-se que teoricamente cada camada só pode comunicar com as camadas adjacentes, o que na prática nem sempre é bem assim.

No caso dos PC's a comunicação é geralmente assegurada por Interrupts.

A definição de interrupts nos PC's não é tão simples como pode parecer à primeira vista pois o microprocessador usado — o 8086 — permite a ocorrência de interrupts tanto por hardware como através da execução duma instrução em Assembler especial (curiosamente chamada INT) sendo a estes chamados interrupts por software.

Cada vez que um interrupt é gerado, é corrida uma rotina para atendimento específico a este interrupt, por exemplo, quando se carrega numa tecla, o teclado origina um interrupt e é chamada uma rotina que depois de ler o caracter vindo do teclado o põe num buffer para posterior tratamento desta ocorrência.

Partindo do centro da cebola, verifica-se que o hardware comunica qualquer facto à BIOS usando um Interrupt por Hardware, isto é, o interesse para as outras camadas, como por exemplo a chegada dum caracter a partir do porto série ou a sinalização de passagem dum certo intervalo de tempo feita pelo Timer.

Depois de gerado este interrupt é função da BIOS efectuar as acções necessárias a cada caso. Esta tarefa muitas vezes é executada com o acesso aos portos destinados a cada periférico.

A partir deste nível todos os interrupts são ocasionados por software.

Assim, para as camadas mais exteriores comunicarem com a BIOS recorrem também a um interrupt. Antes de mencionar como isto é feito torna-se necessário indicar quais os interrupts que são tratados a nível da BIOS bem como a sua função:

INT	00h	Divisão por zero*
INT	01h	Interrupt de Single Step
INT	02h	Interrupt não mascarável*
INT	03h	Interrupt de Breakpoint
INT	04h	Overflow *
INT	05h	Rotina para imprimir o conteúdo do ecrã (Print Screen)
INT	08h	Interrupt do relógio *
INT	09h	Teclado *
INT	0Bh	e 0Ch Interrupts dos portos série *
INT	0Dh	e 0Fh Interrupts dos portos paralelos *
INT	0Eh	Controlador de disco *
INT	10h	Serviços de escrita de ecrã
INT	11h	Devolver o tipo de equipamento usado
INT	12h	Devolve o tamanho da memória em K
INT	13h	Interrupt de acesso a serviços de disco ou disquete
INT	14h	Acesso ao porto série
INT	15h	Interrupt de acesso a srviços especiais
INT	16h	Manipulação do buffer do teclado
INT	17h	Interrupt de acesso à impressora
INT	18h	Carregador de BASIC (nos PC's originais)
INT	19h	Inicialização do Hardware e carregamento do S. O.
INT	1Ah	Serviços para acertar / ler relógio interno
INT	1Bh	Rotina de tratamento do Control-Break
INT	1Ch	Serviços para temporização

NOTA: Os interrupts que estão marcados com um asterisco (*) são accionados pelo Hardware, sendo todos os outros acciados por Software.

Alguns dos interrupts que são accionados por software atrás referidos executam várias funções tornando-se assim necessário explicitar qual a função que se deseja usar. Assim, por exemplo, no caso do INT 10h (16 em decimal) que executa as mais variadas acções de escrita no ecrã, desde limpar o ecrã a escrever um caracter ou mudar para modo gráfico, deve indicar-se num registo da máquina qual a função que se deseja da seguinte maneira:

```
MOV AH, <número da função a usar>
INT 10h
```

Por vezes o que se indica em cima não basta, precisando a BIOS de mais alguns dados para executar uma tarefa sendo esses dados também passados por registos. Se se quiser escrever um caracter deve escrever-se:

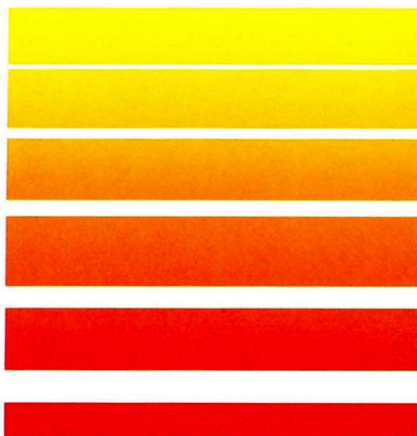
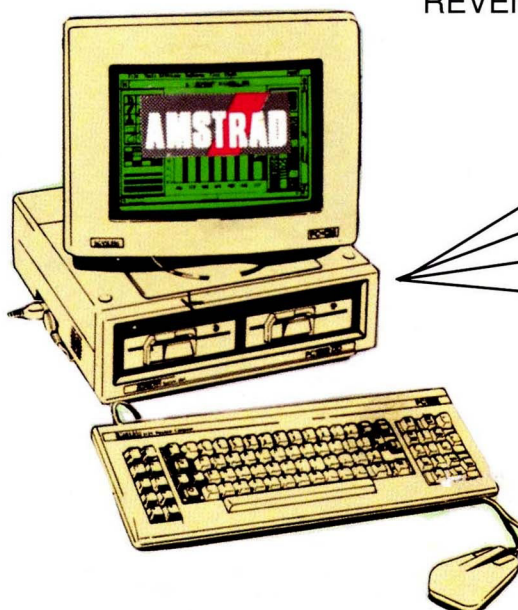
```
MOV AH, 14   ; Função para escrever um caracter
MOV AL, 65   ; Caracter a escrever
MOV BH, 0    ; Página de video onde colocar o caracter
INT  10h     ; Chamar a BIOS
```

A utilização de todas as rotinas da BIOS era por si só motivo para escrever mais uma série de artigos devido à grande quantidade de tarefas que aqui podem ser executadas.

CONCLUSÃO

Nesta primeira abordagem aos sistemas operativos foi dada a definição dum sistema operativo e a maneira de comunicar entre os vários níveis particularizando para o caso dos PC's.

No próximo artigo iremos começar a "sério" a falar do DOS, portanto prepare-se e ponha o seu compilador de C pronto a ser usado...

MIGUEL

BYTE INFORMÁTICA

CONJUNTO MONUMENTAL INFANTE — SALA 204
9000 FUNCHAL

— Porque queremos que a informática chegue a todos

— Compre agora o seu **AMSTRAD** pagando-o em 18 prestações.

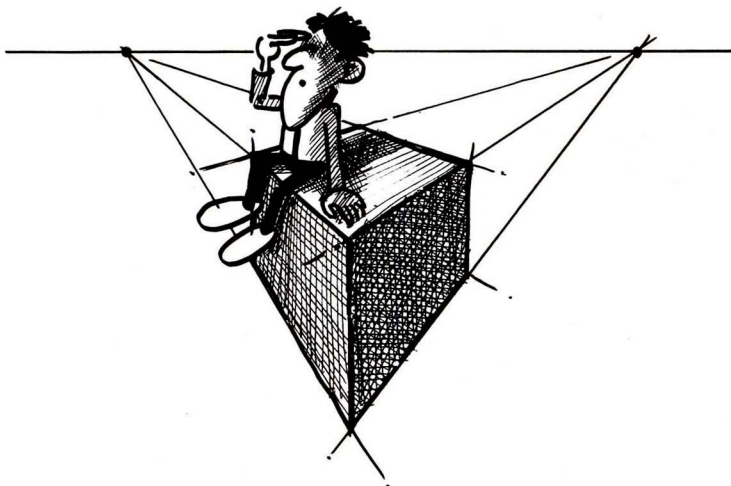
— Prefira o revendedor autorizado **AMSTRAD.**

A RAZÃO DA ESCOLHA CERTA!

— Aplicações por medida.

— Aplicações normalizadas

Contabilidade
Stocks
Facturação
Contas Correntes
Fornecedores/Clientes
Salários
Vídeos

UMA PERSPECTIVA SOBRE A PERSPECTIVA

REPRESENTAÇÃO WIRE-FRAME

Dentre as aplicações que mais facilmente cativam a atenção dos utilizadores estão sem dúvida as gráficas. A verdade é que uma simples esfera a rodar num monitor, provoca mais impacto num distraído visitante de uma feira de informática do que a última palavra em folhas de cálculo ou uma potente aplicação de bases de dados.

Um dos problemas que se põe ao nível da computação gráfica é a representação de objectos em perspectiva. A linha de soluções para este problema é longa, dependendo do grau de sofisticação que se queira dar ao resultado obtido. A mais simples resulta de uma representação wire-frame do objecto. Deste passa a interessar-nos unicamente o conjunto das suas arestas que vão então constituir uma wire-frame, ou em português, uma estrutura de arame.

Vejamos então como escrever um programa que leia a definição de um sólido, de um ficheiro por exemplo, o afixe no monitor em representação wire-frame e permita interactivamente mudar a posição relativa do observador.

ALGUMAS ESTRUTURAS DE DADOS

Um sólido fica definido pelas coordenadas dos seus vértices, e pelo conjunto das suas arestas. Isto corresponde às seguintes estruturas de dados Pascal:

```
Vertices = array [ 1 .. 4,1 .. MAXVERTS ] of real;
```

Cada coluna desta matriz contém as coordenadas homogéneas de um vértice do sólido. Neste tipo de coordenadas é acrescentado um factor de escala às três habituais coordenadas cartesianas x, y, z. Assim, o ponto (a, b, c) é representado pela coluna [x, y, z, w] onde: a = x/w, b = y/w, c = z/w.

```
Par = array [1 .. 2] of integer;
```

Representa um par de vértices, ou seja uma aresta. Assim o par (3,4) é a aresta que liga o vértice representado pela terceira coluna de Vértices ao representado pela quarta.

```
Pares = record
          Desordenado : array [1 ..MAXARESTA] of Par;
          nPares : integer
        end
```

Será o conjunto de arestas que constituem o sólido. É implementado através de um vector acompanhado.

A única coisa que falta é então encapsular tudo isto no tipo Solidos:

```
Solidos = record
            nVertices : integer;
            Objecto : Vertices;
            Imagem : Imagens;
            Aresta : Pares
          end
```

Falta explicar o campo Imagem e nVertices: No primeiro vamos colocar as coordenadas dos vértices no ecrã após as calcularmos claro, e no segundo o número de vértices.

UM MODELO

O passo seguinte é agora arranjar um modelo que nos facilite a conceptualização do problema.

Imaginemos então que temos um cubo e uma máquina fotográfica. Se nos dispusermos a fotografar o cubo, de que dependerá o resultado final? Vai depender da posição relativa entre a máquina e o cubo, e da focagem da lente.

Vamos começar por tratar o primeiro aspecto e para isso nada melhor que considerarmos a existência de um referêncial absoluto O, onde vamos posicionar o cubo e a câmara. Para facilitar vamos supôr ainda que o cubo se encontra na origem deste referêncial.

Seguidamente consideremos a existência de um outro referêncial L em cuja origem vamos posicionar a câmara. A objectiva da câmara estará sempre disposta ao longo do eixo

yL que por seu lado apontará para a origem do referencial absoluto. O eixo zL define a verticalidade da câmara.

TRANSFORMAÇÕES

O nosso problema está pois localizado e consiste em arranjar uma transformação que nos leve do referêncial absoluto ao referêncial da lente L. A posição deste será definida em termos de coordenadas esféricas Teta, Fi, Ro (fig.1). A verticalidade da câmara será dada por um ângulo de Orientação. Com todos estes dados podemos já escrever a transformação:

rot(z, Pi / 2 — Teta) • rot(x, Pi / 2 — Fi) • rot(y, Orientação) • trans(0, Ro, 0) • rot(z, Pi)

Se estava indeciso quanto a ler este artigo, de certeza que depois de olhar para esta expressão deixou de estar. No entanto peço-lhe que continue e desde já prometo que nada de tão complicado aparecerá até à última linha.

Passemos então à explicação. Como já deve ter adivinhado rot significa rotação e trans, translacção. O significado de rot(z, Pi / 2 — Teta) é pois rodar em torno do eixo z de um ângulo Pi /2 — Teta, e trans(0, Ro, 0) significa fazer uma translacção segundo o eixo y de Ro. A explicação é agora bastante intuitiva se pensarmos nas cinco transformações como uma sequência de movimentos que nos levam do referencial absoluto ao referencial da lente. Imaginemos que L começa por estar coincidente com O. Com o primeiro movimento, yL passa a fazer um ângulo Teta com xO. O segundo leva yL a fazer um ângulo Fi com zO enquanto que a sua projecção sobre o plano xyO continua a fazer um ângulo Teta com xO. A terceira rotação destina-se a dar à lente a verticalidade pretendida. Finalmente desloca-se L de Ro ao longo de yL por forma a que os referenciais fiquem à distância Ro pedida. A rotação final destina-se a apontar yL no sentido da origem de O.

TRANSFORMAÇÕES E MATRIZES

Estas transformações representam-se facilmente através de matrizes. Uma translacção seguida de uma rotação é dada pela seguinte matriz:

ex	jx	kx	x
ey	jy	ky	y
ez	jz	kz	z
0	0	0	1

Isto sugere a seguinte estrutura de dados:

TransfMat = array [1 ..4,1 ..4] of real;

onde x, y, z são respectivamente translacções de magnitude x, y, z ao longo dos eixos x, y, z e (ex, ey, ez), (jx, jy, jz), (kx, ky, kz) são os vectores unitários dos eixos coordenados após a rotação.

Por fim, qualquer referêncial será sermpre representado pela transformação que leva do referêncial absoluto ao referêncial em causa. O referêncial absoluto será pois o elemento neutro das transformações, ou seja a matriz identidade.

OS PRIMEIROS PROCEDIMENTOS

Vejamos então como traduzir estas ideias em Pascal.

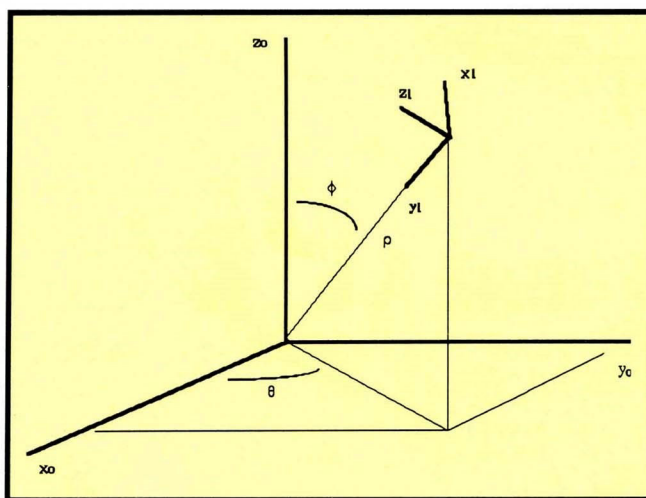



Figura 1

Comecemos por escrever um procedimento que roda um referêncial MatIn em torno do eixo dos x de Teta para produzir um referêncial MatOut:

```
procedure xRot (var MatIn : TransfMat; Teta : real;
                          var MatOut : Transfmat);
var CoSeno, Seno : real;
      Mat : TransfMat;
begin         Seno := sin (Teta);
              CoSeno :=cos (Tata);
              Mat := IDENTIDADE;
              Mat [2,2] := CoSeno;
              Mat [2,3] := -Seno;
              Mat [3,2] := Seno;
              Mat [3,3] := CoSeno;
              Composição (MatIn, Mat, MatOut)
end;
```

Identidade é a matriz identidade. O procedimento Composição limita-se a multiplicar as matrizes e tem a seguinte forma:

```
procedure Composicao (var Transf1, Transf2, Transf :
Transfmat);
var Acum : real;
     m, n, p : integer;
begin for m := 1 o 4 do
     for p := 1 to 4 do
     begin Acum := 0;
          for n := 1 to 4 do
               Acum := Acum + Transf1 [m,n]*
                    Transf2 [n,p];
               Transf [m,p] := Acum
     end
end;
```

O FOCO

O nosso segundo problemas tinha que ver com o foco da lente e com a forma pela qual a imagem se vai afinal produzir na película. Não entrando em pormenores do domínio da óptica, o que acontece é que a imagem de cada ponto do cubo se vai refractar na lente e produzir uma imagem virtual atrás desta. O que vamos ver na película será então uma projecção

dessa imagem segundo o plano da primeira. O cálculo da imagem de um ponto faz-se através de uma transformação Tpersp. Se forem (x, y, z, w) as coordenadas de um ponto no referêncial da câmara L, a sua imagem será:

1	0	0	0	x
0	1	0	0	y
0	0	1	0	z
0	-1/f	0	1	w

onde f é o valor do foco da lente.

ACENTANDO IDEIAS

Temos pois o nosso problema praticamente resolvido. Mas antes de nos atirarmos à programação, vamos acentar algumas ideias.

Após a leitura do ficheiro de entrada sabemos quais são as coordenadas dos vértices do cubo no referêncial absoluto. Ora o que nos interessa é ter as coordenadas no referêncial da lente para então as multiplicarmos por Tpersp e obtermos as coordenadas das imagens virtuais. Mas isto agora já é fácil uma vez que temos uma transformação L que nos leva do referêncial absoluto ao referêncial da lente. As coordenadas dos vértices no referêncial da lente serão pois: Inv (L) . Solido onde Inv(L) representa a transformação inversa de L e solido é do tipo Solidos e contém as coordenadas dos vértices do sólido. As imagens serão pois dadas por: Tpersp . Inv(L) . solido.

Uma vez que vamos precisar de inverter uma transformação, o melhor é escrevermos já um procedimento que cálcula a inversa da respectiva matriz. Verifica-se facilmente que a inversa da transformação:

nx	ox	ax	px		nx	ny	nz	-p•n
ny	oy	ay	py		ox	oy	oz	-p•o
nz	oz	az	pz	é	ax	ay	az	-p•a
0	0	0	1		0	0	0	1

com p = [px, py, pz], n = [nx, ny, nz], a = [ax, ay, az] e ., o produto interno entre vectores.
É agora fácil passar ao código:

```
procedure InvTrans (var MatIn, MatOut : TransMat);
var i,j : integer;
    Acum : real;
begin Transp (MatIn, MatOut);
    for i := 1 to 3 do
    begi Acum := 0;
        for j := 1 to 3 do
            Acum := Acum + MatOut [i,j]*MatOut [4,j];
                MatOut [i,4] :=-1.0*Acum
    end;
    for i := 1 to 3 do MatOut [3] [i] := 0
end;
```

onde Transp cálcula a transposta de uma matriz. É que se olharmos para as matrizes, vemos que à excepção da quarta coluna e da quarta linha a segunda matriz é a transposta da primeira.

O que temos neste momento é a representação do sólido após este atravessar a lente. Falta-nos agora obter uma projecção segundo um plano. O mais fácil é projectá-lo segundo o plano xzL já que para tal basta zerarmos as coordenadas em y obtidas anteriormente. O que se faz na prática é ignorar esta coordenada. Isto é feito com o procedimento ImagemSolido:

```
procedure ImagemSolido (var Transf : TransfMat);
var m, n : integer;
    AuxMat : Vertices;
    Linha : array [1 .. 2] of integer;
begin Linha [1] := X;
      Linha [2] := Z;
        Produto (Transf, Solido.Objecto, AuxMat,
          Solido.nVertices);
    for m := 1 to 2 do
        for n := 1 to solido.nVertices do
            Solido.Imagem [m, n] :=
            AuxMat [linha [m] , n] / Auxmat [W] [n]
end;
```

O procedimento Produto não faz mais que calcular o produto de uma transformação por uma matriz de pontos.

INTERACTIVIDADE...

Até agora não discutimos o carácter interactivo do programa. Como conseguir mudar a posição relativa da câmara? A solução será guardar sempre o resultado de

rot(z, Pi /2 — Teta) • rot(x, Pi / 2 — Fi) • rot(y, Orientação).

Agora se quisermos rodar em torno do objecto de um determinado ângulo Alfa segundo a coordenada esférica Teta não temos mais que aplicar à transformação acima, rot(z, Alfa). No fim haverá sempre que multiplicar o resultado por trans(0, Ro, 0) . rot(z, Pi). Através de um processo análogo solucionavam-se os outros movimentos.

Tudo isto acaba por sugerir a estrutura de dados para a lente:

```
Lentes = record
        Foco, Teta, Fi, Ro, Orientacao : real;
        invTlente, Tpersp, Trans : Transfmat
    end;
```

onde Tpersp vai guardar a transformação Tpersp, Trans a transformação rot(z, Pi / 2 — Teta) . rot(x, Pi / 2 — fi) . rot(y, Orientação) e InvTlente, Inv(L).

Do ponto de vista do utilizador ser-lhe-à fornecido um conjunto de teclas que lhe permitirá movimentar-se em torno do objecto e ainda modificar o valor de Ro e do Foco da lente. Os caracteres correspodentes estão definidos no início do texto do programa apresentado. O que o programa se tem de encarregar é de ler o teclado e reafixar a imagem após calcular as transformações necessárias. A função que se encarrega de fazer isto é Executa que tem como argumento o caracter teclado.

O FICHEIRO DE ENTRADA

A única coisa que falta agora é convencionar o formato do ficheiro de input. Podemos assumir que tem na primeira linha os valores de Teta, Fi, Ro e Orientação iniciais. Na segunda linha virá o foco. Para facilitar as coisas, admitamos que na terceira está o número de vértices, seguido dos respectivos. Haverá um vértice por linha sendo a ordem das coordenadas x, y, z. A seguir virá o número de arestas e por fim as arestas, uma por linha.

Declarando o ficheiro do tipo text, a leitura fica trivial:

```
procedue LeFic;
var i, j : integer;
     InFic : text;
egin assign (InFic, 'base.def');
     reset (InFic);
     with Lente, Solido do  begin
     readln (InFic, Teta, Fi, Ro, Orientacao);
     readln (InFic, Foco);
     for i := 1 to nVertices do
          bengin for j := 1 to 3 do
               read (InFic, Objecto [j, i]);
               readln (Infic);
               Objecto [W, i] := 1
          end;
     readln (InFic, Aresta.nPares);
     for i := 1 to Aresta.nPares do
     begin for j := 1 to 2 do
          read (Infic, Aresta.Desordenado [i] [j];
          readln (InFic);
     end;
     close (InFic)
end             end;
```

A IMPLEMENTAÇÃO

Resta dizer que o programa apresentado foi escrito em Turbo Pascal e tira partido das duas páginas da placa EGA. A ideia é obter alguma animação escrevendo sempre na página escondida e depois fazer o swapping das páginas.

JOÃO HENRIQUE

```
program desenha_solido;

{ Para possibilitar a utilizacao de rotinas graficas e de leitura
  do teclado : }
uses graph, crt;

const  X = 1;
       Y = 2;
       Z = 3;
       W = 4;
       DELTA = 0.73;  { Relacao entre a largura e altura dos pixels }
       MAXARESTA = 200;
       MAXVERTS = 200;

       { Definicao das teclas : }
       MAISRO = '1';
       MENOSRO = '!';
       MAISFOCO = '2';
       MENOSFOCO = '@';
       MAISTETA = 'a';
       MENOSTETA = 'A';
       MAISFI = 's';
       MENOSFI = 'S';
       MAISORIENT = 'd';
       MENOSORIENT = 'D';
       FIM = 'q';

       INC = 1000;
       INCROT = 0.1;


type
       TransfMat = array [ 1 .. 4, 1 .. 4 ] of real;
       Lentes =   record
                    Foco, Teta, Fi, Ro, Orientacao : real;
                    invTlente, Tpersp, Trans : TransfMat
                  end;
       Vertices =    array [ 1 .. 4, 1 .. MAXVERTS ] of real;
       Imagens =  array [ 1 .. 2, 1 .. MAXVERTS ] of real;
       Par =      array [ 1 .. 2 ] of integer;
       Pares =    record
                    desordenado : array [ 1 .. MAXARESTA ] of Par;
                    nPares : integer
                  end;
       Solidos =  record
                    nVertices : integer;
                    Objecto : Vertices;
                    Imagem : Imagens;
                    Aresta : Pares
                  end;

var     Solido : Solidos;
    Lente : Lentes;
    Ordem : char;
    Pagina : boolean;
   IDENTIDADE : Transfmat;

(*************************************************************)
(* Procedimentos de manipulacao de matrizes                  *)
(*************************************************************)

procedure Produto ( var Transf : TransfMat; var MatIn, MatOut : Vertices;
                    nVers : integer );
var Acum : real;
    m, n, p : integer;
begin   for m := 1 to 4 do
          for p := 1 to nVers do
          begin  Acum := 0;
                 for n := 1 to 4 do
                    Acum := Acum + Transf [ m , n ] *
                            MatIn [ n , p ];
                 MatOut [ m, p ] := Acum
          end
end;

procedure Composicao ( var Transf1, Transf2, Transf : Transfmat );
var Acum : real;
    m, n, p : integer;
begin   for m := 1 to 4 do
          for p := 1 to 4 do
          begin  Acum := 0;
                 for n := 1 to 4 do
                    Acum := Acum + Transf1 [ m , n ] *
                            Transf2 [ n , p ];
                 Transf [ m, p ] := Acum
          end
end;

procedure InicIdentidade;
var i, j : integer;
begin for i := 1 to 4 do
        for j := 1 to 4 do
           IDENTIDADE [ i, j ] := ord ( i = j )
end;

procedure Transp ( var MatIn, MatOut : TransfMat );
var m, n : integer;
begin   for m := 1 to 4 do
          for n := 1 to 4 do
             MatOut [ m, n ] := MatIn [ n, m ]
end;

procedure InvTrans ( var MatIn, MatOut : TransfMat );
var i, j : integer;
    Acum : real;
begin   Transp ( MatIn, MatOut );
        for i := 1 to 3 do
        begin  Acum := 0;
               for j := 1 to 3 do
                  Acum := Acum + MatOut [ i, j ] * MatOut [ 4, j ];
               MatOut [ i, 4 ] := -1.0 * Acum
        end;
        for i := 1 to 3 do MatOut [ 4 ] [ i ] := 0
end;

procedure xRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 2, 2 ] := CoSeno;
        Mat [ 2, 3 ] := -Seno;
        Mat [ 3, 2 ] := Seno;
        Mat [ 3, 3 ] := CoSeno;
        Composicao ( MatIn, Mat, MatOut )
end;

procedure yRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 1, 1 ] := CoSeno;
        Mat [ 3, 1 ] := -Seno;
        Mat [ 1, 3 ] := Seno;
        Mat [ 3, 3 ] := CoSeno;
        Composicao ( MatIn, Mat, MatOut )
end;


procedure zRot ( var MatIn : TransfMat; Teta : real;
                 var MatOut : TransfMat );
var CoSeno, Seno : real;
    Mat : TransfMat;
begin   Seno := sin ( Teta );
        CoSeno := cos ( Teta );
        Mat := IDENTIDADE;
        Mat [ 1, 1 ] := CoSeno;
```

Reprodução de gravura a água-forte de José Faria

DDB NEEDHAM & GUERREIRO

AUDIÇÃO

Através da audição, o Homem capta
o sentido das coisas.
E, assim, a realidade vibra mais intensamente,
na aguda percepção dos problemas, nas graves
exigências do dia a dia.

A Sopsi tem da informática uma concepção humanista,
onde o plano do sensível é preponderante.
Por isso, sabe ouvir os seus Clientes. Ausculta as suas
opiniões. Dá ouvidos às suas necessidades.

Hoje, a Sopsi não se limita a representar
e comercializar algumas das melhores marcas
e produtos do mercado, nomeadamente, computadores
pessoais Amstrad e computadores Forum.
Presta, também, a mais eficiente assistência técnica,
como garantia da melhor relação de confiança mútua.
E vai criar lojas de características especiais em vários
pontos do País, para manter com os seus Clientes
um diálogo cada vez mais vivo.

O Sentido da Informática

Sopsi-SOCIEDADE PORTUGUESA DE SISTEMAS DE INFORMÁTICA, S. A.
AV. DA BOAVISTA, 2881-2.º — 4100 PORTO

```
        Mat [ 1, 2 ] := -Seno;
        Mat [ 2, 1 ] := Seno;
        Mat [ 2, 2 ] := CoSeno;
        Composicao ( MatIn, Mat, MatOut )
end;

procedure ImagemSolido ( var Transf : TransfMat );
var m, n : integer;
    AuxMat : Vertices;
    Linha : array [ 1 .. 2 ] of integer;
begin   Linha [ 1 ] := X;
        Linha [ 2 ] := Z;
        Produto ( Transf, Solido.Objecto, AuxMat, Solido.nVertices );
        for m := 1 to 2 do
            for n := 1 to Solido.nVertices do
                Solido.Imagem [ m , n ] :=
                    AuxMat [ linha [ m ] , n ] / AuxMat [ W ] [ n ]
end;

(*********************************************************************)
(* Procedimentos de leitura do ficheiro                              *)
(*********************************************************************)

procedure InicSol;
var i, j : integer;
    InFic : text;
begin   assign ( InFic, 'base.def' );
        reset ( InFic );
        with Lente, Solido do   begin
        readln( InFic, Teta, Fi, Ro, Orientacao );
        readln ( InFic, Foco );
        readln( InFic, nVertices );
        for i := 1 to nVertices do
            begin   for j := 1 to 3 do
                        read ( InFic, Objecto [ j, i ] );
                    readln ( InFic );
                    Objecto [ W, i ] := 1
            end;
        readln ( InFic, Aresta.nPares );
        for i := 1 to Aresta.nPares do
            begin   for j := 1 to 2 do
                        read ( InFic, Aresta.Desordenado [ i ] [ j ] );
                    readln ( InFic )
            end;
      close ( InFic )
end                 end;

(*********************************************************************)
(* Procedimentos que implementam a perspectiva                       *)
(*********************************************************************)

procedure Tpersp;
begin   Lente.Tpersp := IDENTIDADE;
        Lente.Tpersp [ 4 ] [ 2 ] := -1 / lente.foco
end;

{ Inicializa o campo Trans de Lente }
procedure InicTransLente;
var Aux : TransfMat;
begin   Aux := IDENTIDADE;
        zRot ( Aux, PI / 2 - Lente.Teta, Lente.Trans );
        xRot ( Lente.Trans, PI / 2 - Lente.Fi, Aux );
        yRot ( Aux, Lente.Orientacao, Lente.Trans )
end;

{ Sabido o campo Trans de Lente, calcula o referencial da lente }
procedure CalcRefLente ( var Transf : TransfMat );
var Aux : TransfMat;
begin   Aux := IDENTIDADE;
        Aux [ 1 , 1 ] := -1;
        Aux [ 2 , 2 ] := -1;
        Aux [ 2 , 4 ] := Lente.Ro;
        Composicao ( Lente.Trans, Aux, Transf )
end;

{ Apos calcular o referencial da lente, inverte a transformacao
  correspondente }
procedure InvTLente;
var     Aux : TransfMat;
begin   CalcRefLente ( Aux );
        InvTrans ( Aux, Lente.InvTlente )
end;

procedure CalcImagem;
var Produto : TransfMat;
begin   Composicao ( Lente.Tpersp, Lente.InvTlente, Produto );
        ImagemSolido ( Produto )
end;

procedure InicCalc;
begin   Tpersp;
        InicTransLente;
        InvTlente;
        CalcImagem
end;

procedure ModifFiLente ( Ang : real );
var Aux : TransfMat;
begin   xRot ( Lente.Trans, Ang, Aux );
        Lente.Trans := Aux;
        Lente.Fi := Lente.Fi + Ang
end;

procedure ModifTetaLente ( Ang : real );
var Aux : TransfMat;
begin   zRot ( Lente.Trans, Ang, Aux );
        Lente.Trans := Aux;
        Lente.Teta := Lente.Teta + Ang
end;

procedure ModifOrientLente ( Ang : real );
var Aux : TransfMat;
begin   yRot ( Lente.Trans, Ang, Aux );
        Lente.Trans := Aux;
        Lente.Orientacao := Lente.Orientacao + Ang
end;

(*********************************************************************)
(* Procedimentos de escrita para o ecran                             *)
(*********************************************************************)

procedure InicAfixa;
var gm , gd : integer;
    i : integer;
    j:integer;
begin
    gd:=EGA;
    gm:=EGAHi;
    InitGraph(gd,gm,'');
    for j:=0 to 1 do
      begin SetActivePage(j);
            SetFillStyle( SolidFill , BLUE );
              bar(0,0,56,349);
                bar(583,0,639,349);
              bar(0,0,639,24)
      end;
    setviewport(57,24,582,349,true);
    ClearViewPort;
    SetBKColor( CYAN );
    SetColor ( WHITE );
    Pagina := false
end;

procedure Afixa;
var i : integer;
    x1,y1,x2,y2 : integer;
begin   SetActivePage ( ord ( Pagina ) );
        ClearViewPort;
        with Solido, Solido.Aresta do       begin
        for i := 1 to nPares do
          begin
            x1:=320 - round( Imagem[ 1, Desordenado [ i ] [ 1 ] ] );
            y1:=175 + round( DELTA * Imagem [ 2, Desordenado [ i ] [ 1 ] ] );
            x2:=320 - round( Imagem[ 1, Desordenado [ i ] [ 2 ] ] );
            y2:=175 + round( DELTA * Imagem [ 2, Desordenado [ i ] [ 2 ] ] );
            line(x1, y1, x2, y2 );
        end                     end;
        SetVisualPage ( ord ( Pagina ) );
        Pagina := not Pagina { Para a proxima escrever na outra pagina }
end;

function Executa ( Ordem : char ) : char;
begin
    case Ordem of
        MAISFOCO : begin   Lente.Foco := Lente.Foco + INC;
                           Tpersp
                   end;
        MENOSFOCO :begin   Lente.Foco := Lente.Foco - INC;
                           Tpersp
                   end;
        MAISFI :   ModifFiLente ( INCROT );
        MENOSFI :  ModifFiLente ( -INCROT );
        MAISTETA : ModifTetaLente ( INCROT );
        MENOSTETA :ModifTetaLente ( -INCROT );
        MAISORIENT :ModifOrientLente ( INCROT );
        MENOSORIENT:ModifOrientLente ( -INCROT );
        MAISRO :   Lente.Ro := Lente.Ro + INC;
        MENOSRO :  Lente.Ro := Lente.Ro - INC
    end;
    if Ordem <> FIM then
    begin   if ( Ordem <> MAISFOCO ) and ( ORDEM <> MENOSFOCO ) then
            InvTlente;
            CalcImagem
    end;
    Executa := Ordem
end;

procedure FimGraf;
begin SetActivePage(0);
      SetVisualPage(0);
      CloseGraph
end;

begin
    InicIdentidade;
    InicSol;
    InicCalc;
    InicAfixa;
    Afixa;
    while Executa ( ReadKey ) <> FIM do Afixa;
    FimGraf
end.
```

clube

AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

SPOOLER PARA MS-DOS

O FSPOOL permite-lhe recolher toda a informação destinada à impressora num ficheiro. Inclusivé Print-Screen.
A "listagem" será armazenada em formato TEXTO, sendo possível a posterior edição com qualquer editor de texto.
Este comando pode ser activado e desactivado em qualquer altura. É possível a indicação dum "PATH" na designação do ficheiro.

REF. FS-.143

EMULADOR DE CGA PARA CARTA GRÁFICA HERCULES

REF. FS-.144

PROGRAMAS DISPONíVEIS

VER DESCRIÇÃO NOS NÚMEROS ANTERIORES DA AMSTRAD MAGAZINE

- FS-101 BUGS
- FS-102 PINBALL
- FS-103 PITFALL
- FS-104 POKER MACHINE
- FS-105 PYRAMID
- FS-106 RAIN
- FS-107 ROCKETS
- FS-108 XWING
- FS-109 MAHJONG
- FS-110 MATH PAK
- FS-111 EPISTAT
- FS-112 MAHJONG — para ecrã EGA
- FS-113 ALLMAC
- FS-114 ICON MAKER
- FS-115 ALTAMIRA — editor gráfico
- FS-116 DRAW POKER
- FS-117 PIANO MAN
- FS-118 UTILITÁRIOS PARA ECRÃS EGA
- FS-119 WORLD
- FS-120 MUSIC
- FS-121 PAINT
- FS-122 FXMATRIX
- FS-123 BIORRITMO VERSÃO 3.0
- FS-124 TAROT
- FS-125 BLACK JACK
- FS-126 GIN RUMMY
- FS-127 EDWIN
- FS-128 MONOPOLY
- FS-129 ANSIDRAW
- FS-130 CASIOZ
- FS-131 BIORRITMO PESSOAL
- FS-132 BACCARAT
- FS-133 I`CHING
- FS-134 ANSI-ANIMATOR
- FS-135 MAIL
- FS-136 LABEL
- FS-137 TEMAS MUSICAIS
- FS-138 TWCALC22
- FS-139 ORIGAMI
- FS-140 GAMÃO
- FS-141 PRODIAGS
- FS-142 EMULADOR DE Z80 E CP/M 2.2
- FD-901 STAR-SAK, PC-SIZE, FORGET-IT, PC-PLAN, PC-EMS, PC-MULTI, PC-PITMAN
- FD-902 TRIVIA MACHINE
- FD-903 UTILITÁRIOS PARA O WORDSTAR

GEM GRAPH + GEM DRAW

GEM GRAPH — Com a simples movimentação do rato e premindo apenas um botão, podemos obter gráficos profissionais de alta qualidade: de barras, tipo tarte com ou sem explosão, de símbolos, de linhas ou de mapas. Do tamanho e estilo que você decidir; com texto, cores e fundos de relevo para dar ao seu gráfico um aspecto tridimensional.
Gem Graph é um programa com excelentes qualidades gráficas.

GEM DRAW — Desenhos lineares, artísticos, organigramas, esquemas, etc. Escolha os elementos no menú e dê largas à sua imaginação. GEM DRAW converterá o seu PC num estúdio profissional com 6 tamanhos e tipos de letra, 20 livrarias de gráficos disponíveis, 39 funções de trama, régua, alinhamento, etc. e quando o seu desenho estiver perfeito, obtenha a cópia impressa em papel ou transparência.

PREÇO: 24 900$00 **REF. 302, postal 3**

Software concebido para estar instalado no seu computador, em cima da sua secretária para:

- Cálculos rápidos
- Bloco-notas
- Editor de textos compatível WordStar/Turbo Pascal
- Agenda telefónica
- Planeamento de actividades
- Ligação automática de chamadas telefónicas
- Registo de recados e mensagens
- Pesquisa de códigos ASCII

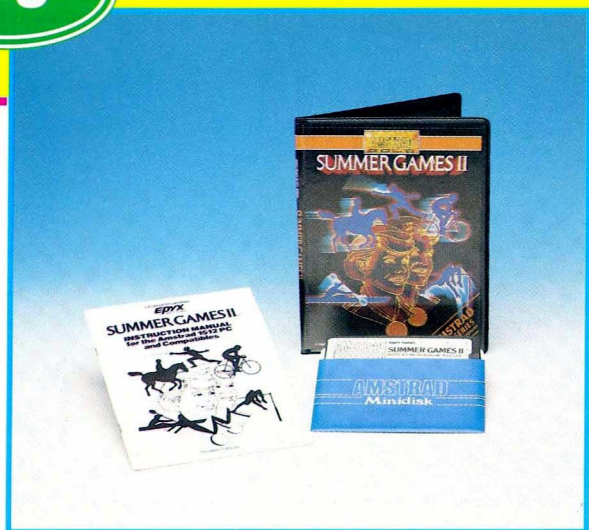
Carregue de manhã o SIDEKICK na memória do computador e fique acompanhado durante todo o dia com esta poderosa ferramenta de trabalho, mesmo utilizando o computador para explorar outro software.

PREÇO: 3 900$00 **REF. 303, postal 3**

MANUAL DE BASIC 2 PARA PC

Ainda não sabe BASIC? Já conhece outro BASIC? Mas não conhece o BASIC 2!
Esta é a linguagem de programação que lhe faz falta conhecer. As sua potencialidades são muitas e convidamo-lo a vir descobri-las.
Através da utilização das janelas do GEM você estabelece um diálogo permanente com a máquina.
O BASIC 2 utiliza, para além de muitas outras particularidades que não encontram nas versões de BASIC disponíveis no mercado, ficheiros indexados próprios das linguagens de gestão. Esta é uma das muitas características que o distingue dos outros. E, concerteza, muito mais.
Este é o manual que lhe faz falta na sua secretária. Não perca a oportunidade de adquirir o manual ao preço... bem... ao preço AMSTRAD.

PREÇO: 2 690$00
REF. 304, postal 3

CLUBE AM

SUMMER GAMES

O quê ?!! O Verão já passou e você não está minimamente interessado no "Summer Games" ?

Está bem, não adquira este jogo e mantenha-se com essa má forma habitual até chegar o próximo Verão. Ao fim e ao cabo nem todos podem andar na praia a mostrar um corpo "Tarzanico".

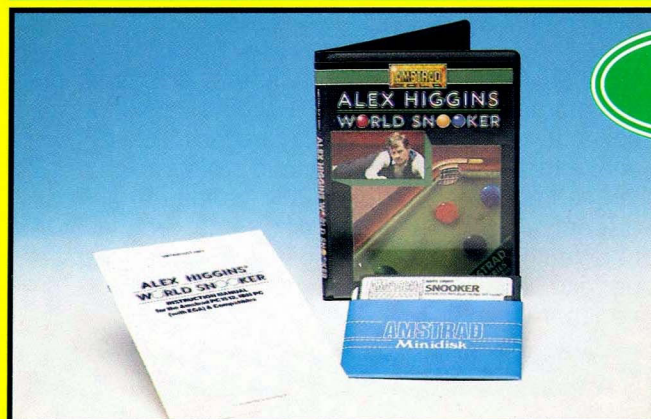
PREÇO: 1 900$00 REF. 327, postal 3

Coberturas para impressora AMSTRAD DMP 3000 e DMP 3160

PREÇO: Elicalfe 2 000$00 REF. 202, postal 3

WINTER GAMES

Não diga que nunca viu neve, ou que só viu gelo dentro de um copo de martini, com este conjunto de jogos de Inverno compilados numa única disquete de 5.25" veja neve, veja gelo, e... se possivel, vá "vendo" uns "martinizitos" enquanto joga.

PREÇO: 1 900$00 REF. 328, postal 3

ALEX HIGGINS SNOOKER

Jogar snooker sempre foi aquilo que a vida lhe deu de melhor. Aliás, uma mesa de snooker no mesmo local onde hoje possui a mesa da sala de jantar, talvez até nem lhe pareça uma ideia assim tão descabida. No fundo, uma mesa é uma mesa, e só é pena que nem todas lhe permitam jogar snooker. Mas... e se em vez de encher a casa com o "hardware" necessário para jogar snooker, você jogasse snooker no ecrã do seu PC Amstrad, acompanhado por ALEX HIGGINS? Pelo menos sempre podia continuar a jantar numa mesa sem buracos e evitava tropeçar nos bocados de giz espalhados pelo chão.

PREÇO: 1 900$00 REF. 329, postal 3

MICROSOFT FLIGHT SIMULATOR version 3.0

Suporta todas as cartas gráficas desde CGA a VGA

Para quem gosta de simuladores de vôo este é **O SIMULADOR DE VÔO**.

Suportando muitas das cartas gráficas habituais nos PC´s inclusivé a Hercules, a EGA, a VGA, e a CGA em visores de cristal liquido ou CRT´s, o Flight Simulator que neste número colocamos à disposição de todos os leitores foi concebido por uma das maiores softhouses da actualidade, senão mesmo a maior - a Microsoft - e é no minimo um simulador excelente a todos os níveis. Em termos de gráficos, por exemplo, para além de suportar as cartas gáficas já referidas e de delas extraír as capacidades que lhes são inactas, suporta ainda outras cartas gráficas não previstas na versão base mas adicionaveis através de drivers externos.

A simulação que pode decorrer num de três aviões diferentes, escolhido pelo utilizador, pode basear-se em operações de descolagem, aterragem, ou vôo normal, sofrendo, ou não, efeitos climatéricos (chuva, vento, neve, etc), ou temporais (dia, fim de tarde, noite, etc.), e estando, ou não, condicionada a um conjunto enorme de outros factores, entre os quais podemos referir os vôos em esquadrilha, ou em perseguição, quer em periodos de paz, quer em periodos de guerra.

O nível de realidade da simulação e controlavel pelo utilizador através de opção acedida por teclado, e para os utilizadores menos à vontade num "cock pit" existe ainda a possibilidade de assistir a lições de vôo sub-divididas por tarefas a executar. A documentação é composta por um enorme manual, diversos mapas, e um pequeno livro de "Quick Reference" (referências rápidas), apoiando de uma forma melhor do que excelente o jogo que se encontra dividido pelas duas disquetes de 5.25" que complementam a package.

Para além do interesse do jogo, pensamos que é digno de nota o facto dele suportar e tirar proveito das cartas VGA, facto que, sem dúvida, o torna único no mercado português.

PREÇO: 9 900$00 **REF. 330, postal 3**

QUICK BASIC versão 3.0

SUPORTA O PROCESSADOR ARITMÉTICO 8087

Uma excelente linguagem de programação e um óptimo compilador de programas concebidos em BASICA ou GW-BASIC, o Quick BASIC proporciona a todos os programadores desta linguagem uma velocidade de processamento que embora não sendo tão grande como a que se obtém no dialecto da mesma linguagem lançado pela Borland, é muito mais standard.

Para todos os utilizadores do GW, o Quick BASIC só pode ser a evolução perfeita. Baseado num set de instruções que quase se pode considerar cem por cento igual ao do dialecto GW, o QB traz-nos toda a velocidade de uma linguagem compilada, as facilidades de "debugging" comuns aos interpretadores da mesma linguagem, e um completo manual de utilização, por um preço impossivelmente baixo!!!

PREÇO: 15 000$00 **REF. 331, postal 3**

DMP 4000

- MANUAL DE UTILIZAÇÃO EM PORTUGUÊS

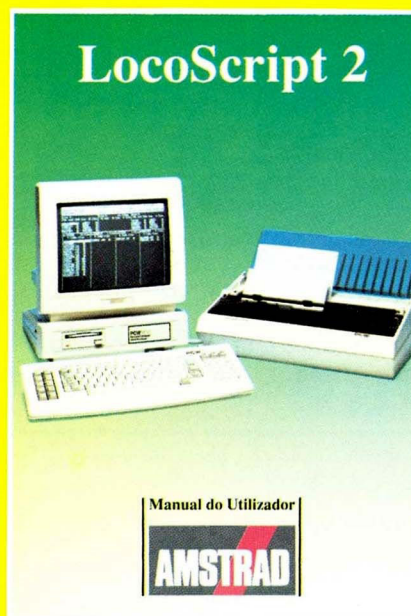
Com uma qualidade de impressão relativamente elevada tendo em consideração que se trata de uma impressora de 9 agulhas, a DMP 4000 pode distinguir-se actualmente como uma impressora bem sucedida no mercado nacional. Tal facto, constituiu uma das razões que nos levou a optar pela inclusão do seu manual de utilização, EM PORTUGUÊS, nesta secção da AM, procurando com isso continuar a proporcionar aos nossos leitores informação tão detalhada quanto possivel, numa linguagem tão simples quanto possivel, a um preço nitidamente impossivel.

PREÇO: 500$00 REF. 320, postal 3

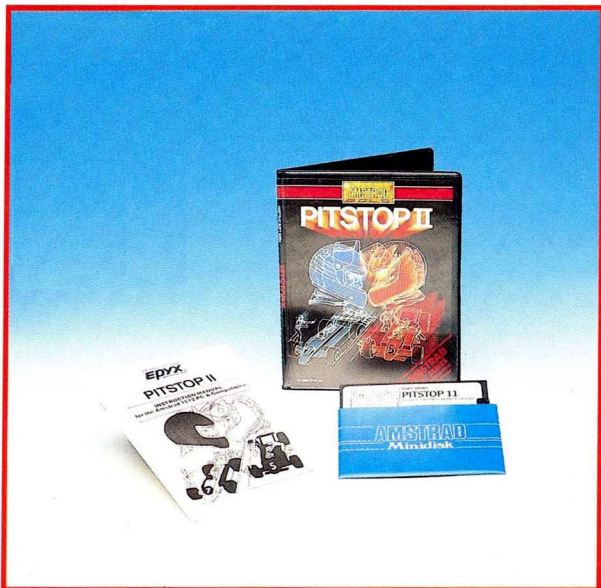
LOCOSCRIPT 2 (para PCW 9512) — Manual do Utilizador EM PORTUGUÊS

Quase quatrocentas páginas de texto, figuras, esquemas, e exemplos, constituem o mais completo livro em português sobre um processador de texto que tem arrastado centenas de pessoas dos teclados das máquinas de escrever para os teclados das modernas máquinas de processamento de texto.

Dividido em quatro partes distintas o manual do Locoscript que aqui apresentamos inicia o seu passeio pelo processador de texto em causa, através de uma passagem pelas "Noções Básicas" e "Refinamentos", concluindo a dissecação do tema com as "Funções Avançadas" disponiveis, e complementando todas estas partes e informações com um detalhado, e bem estruturado, Apêndice, repartido por 5 assuntos diferentes. Tudo o que o utilizador do Locoscript 2 precisa saber para resolver eventuais problemas menos comuns, ou apenas escrever uma simples carta, pode encontrar-se neste manual ao cabo de meia dúzia de segundos de procura.

Utilizando um conhecido slogan há algum tempo passado no pequeno écran podemos mesmo dizer que: se já possui um PCW, utiliza o Locoscript 2 e não possui este manual, DE QUE É QUE ESTÁ À ESPERA ?!!

PREÇO: 1 200$00 REF. 322, postal 3

Coberturas para computador AMSTRAD PC 1512 e PC 1640

PREÇO: Elicalfe 4 530$00 REF. 201, postal 3

PITSTOP II

As corridas de carros nunca foram um desporto para praticar sozinho. PITSTOP II, vem precisamente comprovar essa afirmação ao permitir que dois jogadores possam correr simultâneamente, lutando pelo primeiro lugar. Se, no entanto, não tiver alguém contra quem competir, o seu computador pessoal estará sempre pronto para lhe fazer frente numa dura corrida ao longo de qualquer uma das três pistas possiveis.

O objectivo principal é simples: ganhar a corrida. Contudo para conseguir atingi-lo você terá que tomar várias decisões estratégicas enquanto garante que os seus adversários não o ultrapassam. Os pneus, por exemplo, ou a falta de combustivel, podem obrigá-lo a parar, resta-lhe a si a opção de efectuar as paragens necessárias ou de se arriscar a ficar a meio do caminho. No fundo como, já referimos, entre o primeiro e o último lugar existe apenas uma enorme dose de perícia, um pouco de bom-senso, e uma boa estratégia.

PREÇO: 1 900$00 REF. 323, postal 3

MEAN 18

Gostava de praticar golf mas não tem espaço no local onde vive, e como alternativa, tentar jogar com os buracos que existem na estrada por onde passa diariamente, parece não ser a melhor solução.

Se é esse o seu problema, não se preocupe nem mais um minuto com ele. O MEAN 18 transforma o seu PC num óptimo campo de golf, e deixa-lhe a si a possibilidade de executar magnificas tacadas, ou até mesmo de modelar o campo de acordo com o seu gosto pessoal, evitando a cansativa tarefa de carregar os tacos, ou andar várias centenas de metros atrás da pequena bola branca, que parece bater em todo o lado menos no fundo dos buracos.

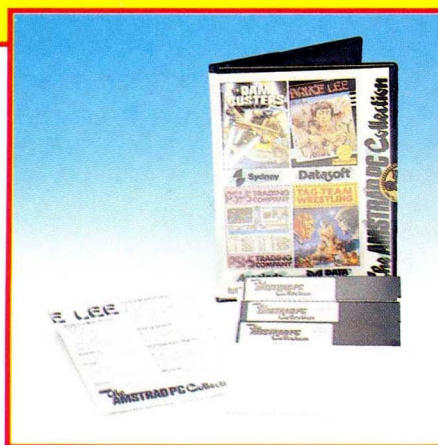
Jogar golf pode ser muito mais fácil!

PREÇO: 1 900$00 REF. 324, postal 3

THE AMSTRAD COLLECTION

Quatro jogos, três disquetes, dois minutos a preencher o postal para os encomendar, uma única oportunidade de adquirir tudo isto por este preço.

Jogos incluidos nesta package:

THE DAM BUSTERS - SYDNEY
BRUCE LEE - DATASOFT
PSI-5 TRADING COMPANY - ACCOLADE
TAG-TEAM WRESTLING - DATA EAST

PREÇO: 1 900$00 REF. 321, postal 3

MICROSOFT WORKS

Descrever o WORKS em tão pouco espaço, seria completamente impossivel, para além de que estariamos apenas a repetir aquilo que a maior parte dos utilizadores já ouviu acerca desta package integrada. No fundo em tão poucas linhas apenas podemos dizer que o WORKS integra quatro poderosas ferramentas prontas para satisfazer a maior parte das necessidades informáticas de qualquer utilizador.

Processador de texto, folha de calculo, e base de dados, são apenas 3 das 4 aplicações integradas nesta package. A quarta aplicação pode funcionar como complemento de cada uma destas ou independente de todas elas, visto que se trata de uma package de comunicações.

A complementar as 12 disquetes fornecidas (8 disquetes em formato 5.25", e 4 com o mesmo conteúdo em formato 3.5") um extenso e completo manual com mais de 600 páginas ordenadas de uma forma lógica, e incluindo um completo, e útil, indice, torna o WORKS a package ideal para quem tem pouco tempo para aprender a "mexer" no computador mas deseja aproveitar todas as suas potencialidades.

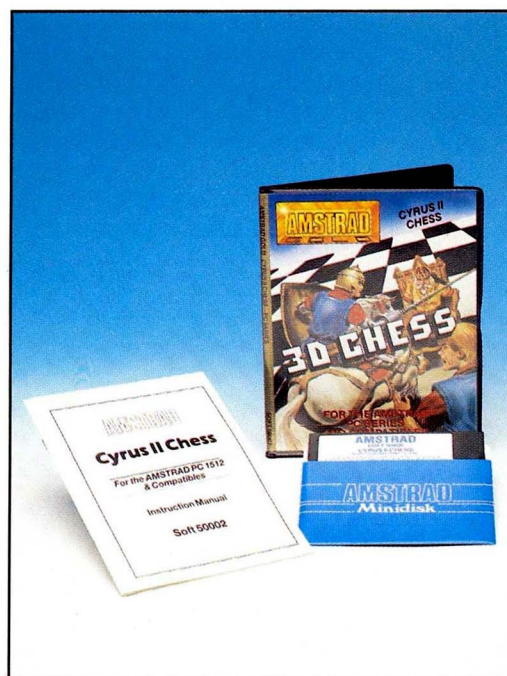
"Um dia de trabalho numa hora de WORKS", podemos afirmar que é a melhor forma de descrever o que esta "pequena maravilha" pode fazer por si. Tudo o resto está dito nas entrelinhas do que dissemos, e demonstrado no software que lhes deu origem.

PREÇO: 37 500$00 REF. 325, postal 3

PC

CYRUS II CHESS

Adquirir o Cyrus II é adquirir um excelente jogo de xadrês com as seguintes caracteristicas:

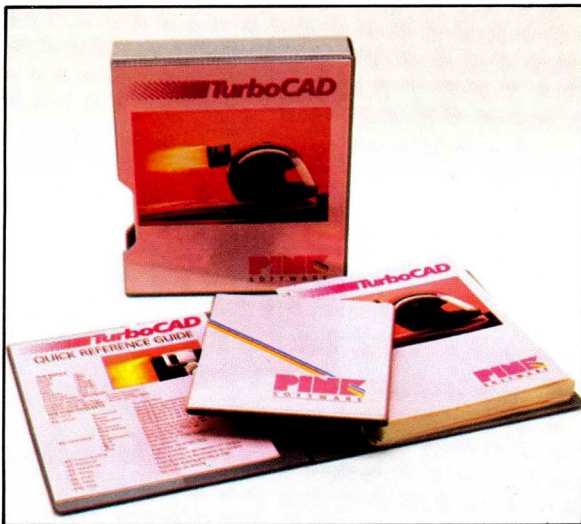
* tabuleiro a duas ou três dimensões, comutaveis a qualquer momento através do teclado;
* suporte de carta gráfica CGA e EGA 16 cores (PC 1640 ECD);
* 16 níveis de dificuldade;
* possibilidade de executar cópias da partida em papel, através de qualquer impressora do tipo da DMP 3000, DMP 4000, ou compatíveis;
* possibilidade de gravar o jogo no meio de uma partida para posteriormente o retomar no mesmo ponto;
* inúmeras opções de análise e ajuda durante a partida;
* um completo manual de 27 páginas;
* um preço de fazer rir.

PREÇO: 1 900$00 REF. 326, postal 3

TODOS OS PREÇOS INCLUEM O TRANSPORTE E O I.V.A. A 17%

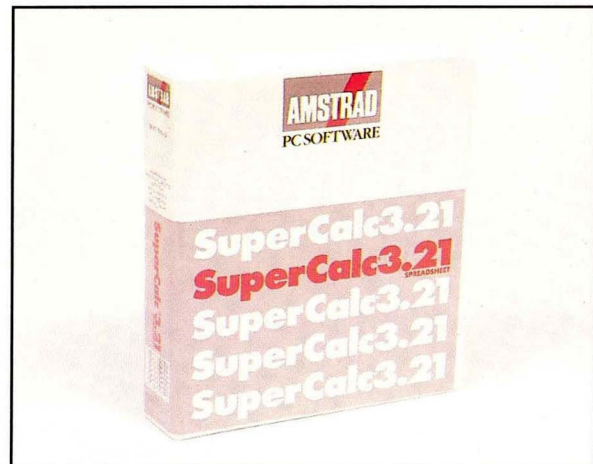
CM1 — CONJUNTO DE 5 JOGOS SORTIDOS PARA CPC

Se é possuidor de um CPC, se tem entre 5 e 95 anos, se tem tempo para jogar e não tem jogos — então tem um grave problema.
Felizmente nós propomos-lhe uma solução.
5 Cassetes com 5 jogos (surpresa) diferentes, vão diverti-lo por muito mais de 5 horas e custar muito menos de 5 contos, embora também custem um pouco mais de 5 escudos.

PREÇO: 990$00 REF.313, postal 3

TurboCAD

De instalação fácil, e utilização simplificada como consequência do funcionamento baseado em menus tipo "pop-up" o TurboCAD pode ser o utilitário que você procura para "dar asas á sua imaginação" no dominio do desenho técnico.
Acompanhado por um completo manual que lhe permite entrar sem grandes dificuldades no mundo do Desenho Assistido por Computador, o TurboCAD assegura a compatibilidade com o AutoCAD (uma das "packages" de CAD mais populares entre os utilizadores de computadores), sendo cerca de 9 ou 10 vezes mais económica do que esta última.

PREÇO: 27 500$00 REF.318, postal 3

SUPERCALC 3.21

O standard em folhas de cálculo é, ainda hoje, nitidamente imposto pelo LOTUS 1,2,3. Ninguém sequer coloca isso em causa. O que começa a colocar-se em causa são as vantagens de utilização desta folha de cálculo numa altura em que existem dezenas de outros utilitários com o mesmo fim, compativeis com o LOTUS, mas... muito mais possantes.
É este, por exemplo, o caso do SuperCalc, agora disponivel na sua versão 3.21.
O SuperCalc foi uma das "packages" que soube tirar proveito do facto de não "rasgar" mercado.
Aproveitando os resultados das experiências dos seus "adversários", o SuperCalc 3.21 melhorou muitas das suas caracteristicas, apresentando por exemplo, entre muitas outras qualidades dignas de nota, modos de representação gráfica superiores aos que a maior parte dos utilitários deste tipo incluem, uma boa velocidade de processamento de dados, e um conjunto de "HELP screens" mais do que suficiente para se começar a tirar proveito da "package", mesmo antes de se começar a ler o detalhado manual que a acompanha.
Conclusão: se nunca utilizou uma folha de cálculo, o SuperCalc 3.21 é-lhe indispensável; se já utiliza uma folha de cálculo o SuperCalc 3.21 é-lhe indispensável.

PREÇO: 19 900$00 REF.319, postal 3

TODOS OS PREÇOS INCLUEM O TRANSPORTE E O I.V.A. A 17%

PC

EXCLUSIVO DO CLUBE DE LEITORES

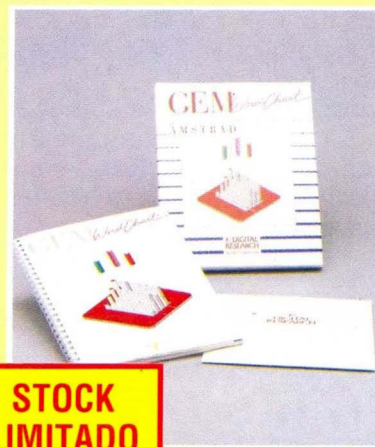
JÁ NÃO PRECISA DE SAíR DE CASA PARA IR JOGAR

POKER

AO CASINO

O jogo Good Luck é uma réplica do popular Poker das máquinas dos casinos, permitindo todo o tipo de jogadas — 2 pares, sequência, fullen, etc. e, para os mais destemidos, dobrar ou perder

PREÇO: 2 000$00 REF.306, postal 3

GEM WORDCHART

STOCK LIMITADO

Actualmente, mais de 80% das apresentações são feitas através de palavras — e não de gráficos.

O GEM WORDCHART, concebido com a intenção de lhe servir de instrumento de trabalho na realização simples de apresentações, permite a utilização de diversos tipos de letras com recurso a inúmeras variantes de cada tipo, selecção de limitadores e formatos, e combinação de cores, através de menus do tipo "drop-down".

Para lhe tornar a composição da folha mais fácil, o texto aparece no écran exactamente igual à posterior cópia impressa, e a largura das colunas pode seleccionar-se com a simples pressão de um botão do "mouse".

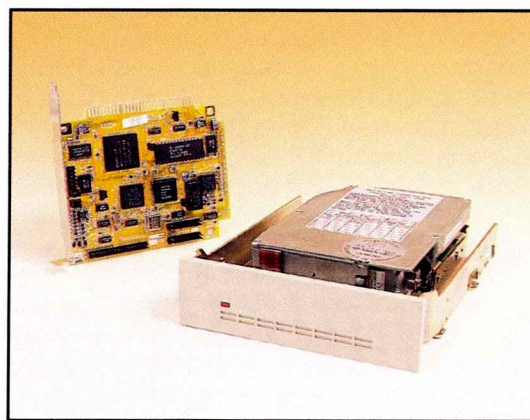
Em resumo, o GEM WORDCHART, situa-se entre o PRINT MASTER e o PAGE MAKER, apresentando no entanto, em relação a um e a outro, algumas vantagens na concretização de pequenos trabalhos.

PREÇO: 9 900$00 REF. 308, postal 3

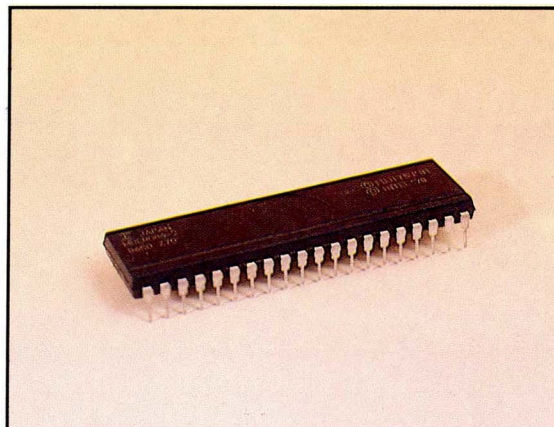
DISCO RIGIDO DE 30 MB

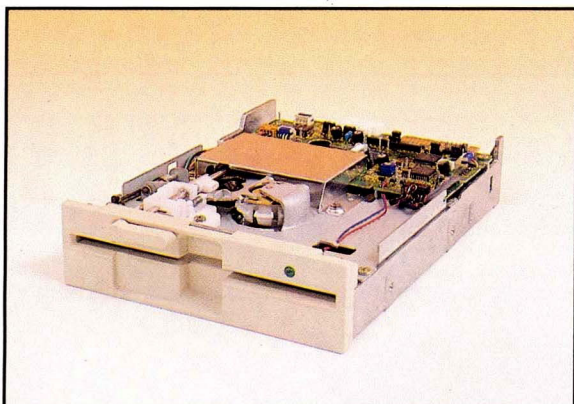
COM VENTOÍNHA E CONTROLADOR

Resultado de uma deficiente análise das necessidades pessoais, de um investimento em meios informáticos necessáriamente baixo, ou de qualquer outra razão menos generalizada, a aquisição de computadores pessoais com uma ou duas drives sempre foi superior à de equipamentos com uma memória de massa de maior capacidade. Consequência desse facto, é quase sempre a posterior troca do equipamento adquirido, ou a incessante procura de um disco rigido com uma capacidade de armazenamento razoável, e um custo "impossivelmente" baixo.

O disco que lhe propomos pode não ser aquele que melhor lhe convem em termos de preço, mas é concerteza uma boa aquisição, se o relacionarmos com os restantes componentes deste tipo já existentes no mercado nacional. De qualidade excelente (diga-se a propósito que o controlador que o acompanha é da Western Digital), este disco será sempre a sua melhor aquisiçãol, se o seu computador pessoal ainda não inclui um semelhante. Ofreça-o a si mesmo, você merece.

PREÇO: 119 000$00 REF. 901, postal 3

PROCESSADOR ARITMÉTICO INTEL 8087 (8MHZ)

Se lhe dissessem que o seu computador pessoal em determinadas situações pode funcionar com uma velocidade de processamento cem por cento superior áquela em que neste momento funciona, estariam sem dúvida a pensar na simples inserção de um processador aritmético na placa principal do seu PC. Tarefa que mesmo uma criança poderá levar a cabo com sucesso a inserção do circuito integrado INTEL 8087 no suporte a ele destinado na placa principal dos PC´s Amstrad, pode com efeito, em certas situações, duplicar a velocidade de processamento da máquina em que está inserido, aumentando-a sempre consideravelmente em todas os outros casos.
Imagine, por exemplo, a velocidade que a sua aplicação em Turbo BASIC, Turbo Pascal, ou Turbo C (para não citar muitas outras) pode atingir com a adição de um simples integrado ao hardware já disponivel, isto para não falar das aplicações de CAD que costuma utilizar, ou de todas as outras aplicações "pesadas" que entretanto recusou por "trabalharem a vapor" numa máquina da era nuclear.

PREÇO: 54 000$00 REF. 902, postal 3

DRIVES DE 5.25"

Em tempos adquiriu um PC com uma única drive, e agora deseja adicionar-lhe uma segunda drive de 5.25" esta oferta soluciona-lhe o problema. Fácil de instalar com alguma habilidade, e uma dose igual de paciência e tempo livre, esta drive de 5.25" vai poupar-lhe o dinheiro que o técnico lhe leva para proceder a uma instalação deste tipo, proporcionando-lhe muito mais gozo pessoal por no final da operação poder afirmar que foi você quem fez a instalação da drive.
Concluida a instalação, você ganhou mais experiência, e... sobretudo ganhou mais dinheiro.

PREÇO: 15 000$00 REF. 903, postal 3

FITAS PARA IMPRESSORA

Por muito boa que seja uma impressora, mais tarde ou mais cedo ela acaba sempre por nos aborrecer. Talvez no futuro as impressoras consigam produzir automáticamente os seus próprios consumiveis, mas por agora somos nós que penosamente os temos de adquirir. As fitas para impressora, nitidamente inseridas nesta categoria, possuem um preço cada vez mais elevado e, apesar disso, são muitas vezes dificeis de encontrar na loja onde costumamos fazer as nossas compras informáticas. Por esta razão nada melhor do que comprar as fitas de que necessita, quando necessita, sem sequer ter de se preocupar em encontrá-las, ou mesmo ter de se deslocar para adquiri-las.

DMP 2/3/3160	1 400$00	REF. 904	postal 3
DMP 4000	2 100$00	REF. 905	
LQ 3500/PCW	1 400$00	REF. 906	
LQ 5000	2 650$00	REF. 907	

DISKETTES AMSTRAD

Em 3", 3.5", ou 5.25" as diskettes Amstrad são fornecidas em conjuntos de 10 unidades com caixa plástica, garantindo uma perfeita formatação e fiabilidade dos dados armazenados.

3"	PREÇO: 8 490$00	REF. 315	postal 3
3.5"	PREÇO: 5 990$00	REF. 316	postal 3
5.25"	PREÇO: 2 690$00	REF. 317	postal 3

MANUAL DO PC EM PORTUGUÊS

Será que os computadores só podem ser utilizados por quem sabe inglês?

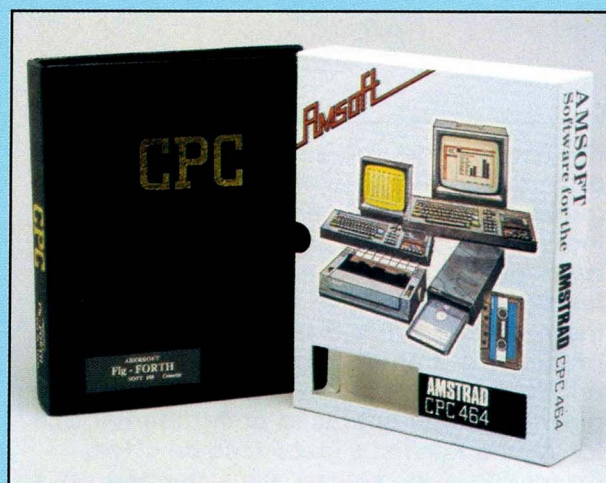
É evidente que não. Embora o conhecimento da língua inglesa facilite a aprendizagem, nunca se poderá considerar indispensável para este efeito. No nosso país, são cada vez mais frequentes as marcas que traduzem os manuais e as packages, e adaptam os teclados, para poderem possuir boas soluções informáticas em mercados que nada têm a ver com a língua inglesa.

Foi assim, seguindo esse princípio, que AM optou por incluir nesta secção a tradução do MANUAL DO PC, para facilitar a vida a todos os que em Portugal preferem ler em português.

PREÇO: 1 900$00 REF. 310, postal 3

FORTH P/ CPC

Num momento em que começam a surgir no mercado alguns processadores que possuem como linguagem "natural" o FORTH, torna-se interessante poder oferecer aos possuidores dos CPC a hipótese de experimentar o poder desta linguagem como forma de comunicar com a máquina. Com algumas vantagens sobre o BASIC (nomeadamente uma maior velocidade de processamento), o FORTH continua a manter inúmeros adeptos entre os programadores e utilizadores de computadores, que não hesitam em defendê-lo, em muitas situações, como uma das melhores linguagens de programação.

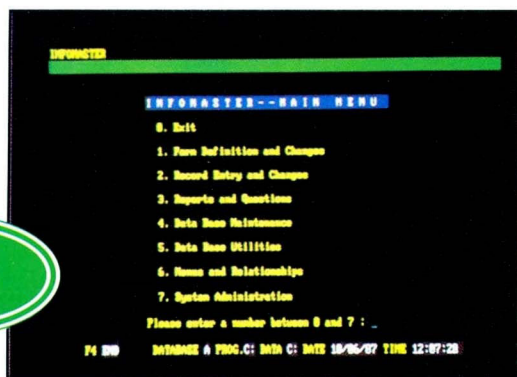
APRESENTADO EM CASSETE

PREÇO: 900$00 REF.314, postal 3

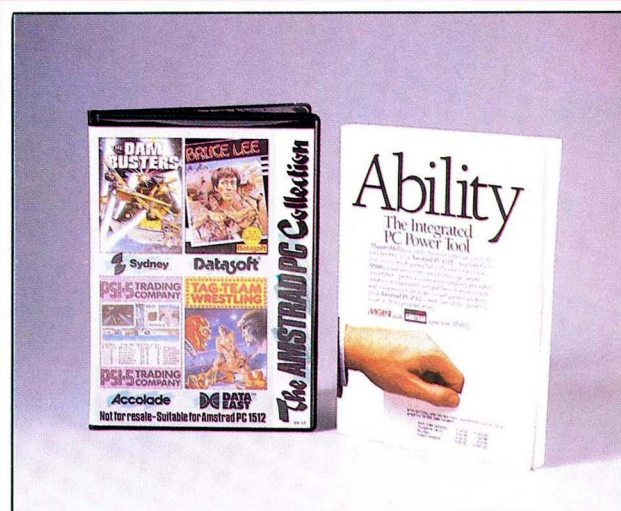
MANUAIS TÉCNICOS DO PC 1512 E PC 1640

Para a maior parte dos utilizadores dos computadores pessoais Amstrad, a informação contida na documentação que acompanha o PC, é mais do que suficiente para conseguir tirar dele todo o proveito que sempre se visou como objectivo desde o momento da sua compra. Para alguns outros, no entanto, as necessidades são um pouco diferentes e para melhor poderem manipular a máquina com que habitualmente trabalham, um pouco mais de informação é, no mínimo, desejável.
Os manuais técnicos do PC 1512 e PC 1640 incluem mais informação sobre a sua máquina do que aquela que provavelmente irá necessitar. Vão, portanto, além da referida "informação desejável". No fundo, como a própria designação deixa entender estamos a falar de manuais técnicos, bastante técnicos, para técnicos, ou futuros... técnicos.

PC 1512 PREÇO: 7 700$00 REF. 908, postal 3
PC 1640 PREÇO: 7 700$00 REF. 909, postal 3

INFOMASTER

PC

Em bases de dados, é verdade que o DBASE criou um standard, mas não é menos verdade que INFOMASTER ultrapassou esse standard.
Permitindo uma utilização eficiente após alguns minutos de trabalho, possibilitando a utilização de um máximo de 65535 registos em cada ficheiro, e um máximo de 255 campos em cada registo, o INFOMASTER torna-se o sistema de gestão de base de dados mais adequado para as pequenas empresas.
Funcionando num sistema de menus que permite a fácil manipulação de informação, e a configuração da base de dados por utilizadores com um mínimo de conhecimentos, esta package utiliza parte da RAM como cache, conseguindo deste modo uma velocidade que em determinadas situações se pode considerar cerca de 400% superior à das bases de dados convencionais.

PREÇO: 17 900$00 REF. 311, postal 3

ABILITY + 4 JOGOS

Package integraao de programas que lhe oferece:

a) Base de Dados.
b) Folha de Cálculo.
c) Gráficos de Gestão.
d) Processamento de Texto.
e) Comunicações.
f) Gerador de Apresentações.

PC

Incluindo:

1) Manual de fácil leitura e manuseamento.
2) Utilização compartilhada de dados para as diferentes aplicações.
3) Integração activa entre os programas, (não realizável em programas conhecidos do mercado).
4) Com o programa APRESENTAÇÃO, incluído no Ability, podem preparar-se informações obtidas com os dados manuseados com o programa base.

E ainda 4 Jogos: "The Dam Busters", "Bruce Lee", "Psi 5 Trading Company" e "Tag Team Wrestling".

PREÇO: 8 900$00 REF.301, postal 3

WINDOWS:

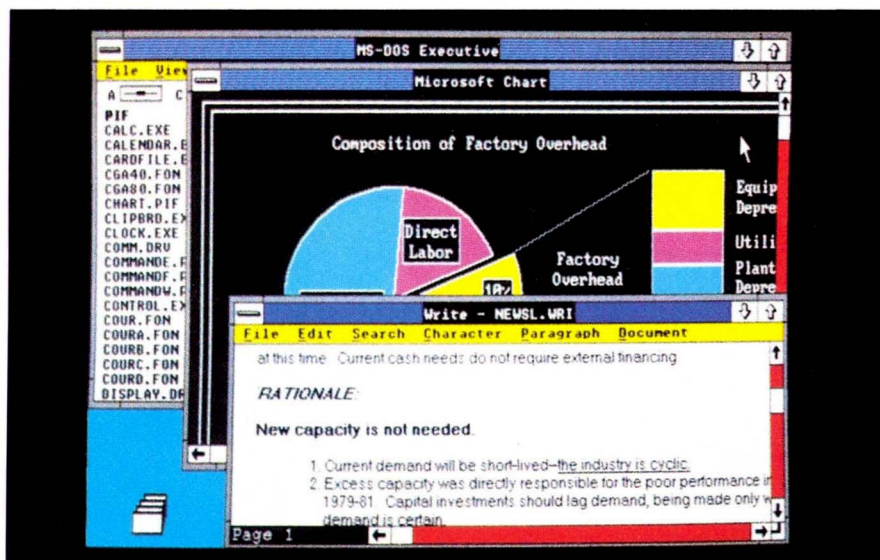
NOVA VERSÃO OU APENAS UM PEQUENO UPGRADE?

INTRUDUÇÃO — BREVE VISÃO GERAL SOBRE O WINDOWS

Cada vez mais se verifica, hoje em dia, que para que um programa tenha sucesso, é necessário que o utilizador se sinta confortável a utilizá-lo. O Windows verifica esta condição e todos os programas que correm sobre ele se apresentam da mesma forma. Tomemos para exemplo, o Pagemaker.

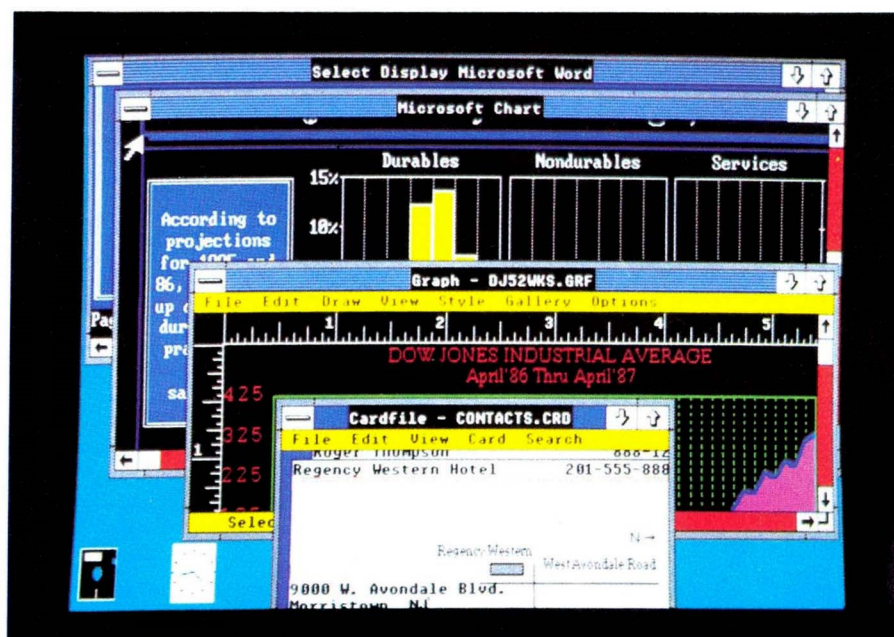
Contudo, já lá vai algum tempo desde que se começou a falar em MicroSoft Windows e..., se se considera o DOS como o ambiente de utilização mais falado nos últimos tempos, o Windows pode mesmo ser o segundo, senão pelo menos o mais discutido. E isto porque a sua primeira versão para além de ter revolucionado o mundo da utilização em DOS (por ser extraordinariamente user-friendly) era, apesar de tudo, muito lenta.

Mas porquê o Windows?

O Windows providência para além do mais, um importante auxiliar na utilização do computador, não só para utilizadores inexperientes que deixam de ter de saber a sintaxe dos comandos do DOS (o que lhes aumenta extraordinariamente a curva de aprendizagem), mas também para utilizadores avançados e mesmo programadores que passam a ter à disposição uma série de pequenas aplicações, tais como a calculadora, o calendário, o bloco de notas, o relógio, etc., que embora não sejam residentes, podem ser facilmente chamadas ou acedidas pelo simples carregar de um botão do mouse. E para trabalhos mais específicos dispomos ainda de um processador de texto (write), de um programa de pintura ou desenho (Paint), de um spooler e de um programa de comunicação, para além de um pequeno jogo para diversão nos intervalos do trabalho. Em minha opinião, melhor do que tudo é a capacidade de intercomunicação entre as aplicações, visto ser sempre possível passar informação de uma aplicação para outra através do ClipBoard. É ainda possível pôr a funcionar programas neste ambiente que não tenham sido escritos propositadamente para ele através dos ficheiros PIF (Program Information File).

CONCLUSÃO: torna-se realmente agradável correr qualquer programa no ambiente Windows desde que se tenha à disposição um computador com boas performances, porque ao tirarmos o máximo proveito do Windows, podemos correr o risco de estar a tornar a sua utilização demasiado pesada para um qualquer computador a 8 MHz, por exemplo. A nova versão do Windows veio já, em parte, remediar este grande problema. Para além de ser mais rápida, tornou ainda mais fácil a utilização do Windows pelo teclado. Mas, vejemos mais em pormenor, esta nova versão e a sua aplicação em ambientes verdadeiramente multitasking: o Windows 386.

NECESSIDADES DE HARDWARE

A nova versão do Windows vem comercializada numa package com 7 disquetes para IBM PCs, XTs, ATs E PS/2 ou compatíveis e em 3 disquetes de alta densidade de 5.25" ou 3 disquetes de 3.5" para computadores baseados no microprocessador Intel 80386 ou compatíveis. Vem acompanhada de um manual de 500 páginas adicionados de 33 páginas específicas para o Windows 386.

Windows 2.03	Windows 386
7 Disquetes de 360Kb	3 Disquetes de Alta densidade de 5.25" e 3 disquetes de 3.5"
Manual Guia do Utilizador com 500 páginas	"Manual Guia do Utilizador" com 500 páginas e 33 páginas de "Utilizando o Microsoft Windows 386"

Em termos de necessidades de Hardware, o novo Windows 2.03 necessita de um IBM PC, XT, AT, PS/2 ou compatível com uma placa gráfica compatível com CGA, Hercules, EGA ou VGA, de duas drives ou uma drive e um disco rígido e de pelo menos 512Kb de RAM (embora sejam recomendados a 640Kb), contra 320Kb (embora recomendados fossem os 512Kb) da versão anterior. Em termos de sistema operativo, a nova versão necessita do DOS 3.0 ou versão superior enquanto que a anterior necessitava apenas do DOS 2.0 ou versão mais recente.

Um facto muito importante é que a nova versão pode ainda usar memória adicional para acelerar a velocidade do sistema evitando, em todo o caso, as demoradas escritas e leituras sobre o disco rígido ou sobre a disquete. E isto porque, um factor determinante a ter em conta na rapidez com que se executa uma aplicação é a quantidade de memória que se tem disponível. Senão vejamos: à medida que vamos pondo programas a funcionar sobre o Windows, é alocada uma determinada quantidade de memória. Quando pretendemos carregar mais uma aplicação e já não há memória suficiente, torna-se necessário descarregar uma aplicação que já lá esteja, para que possa ser aberto novo espaço para a nova aplicação. É assim necessário usar unidades mecânicas, tipo disco, para suportar o programa que foi descarregado, sendo esta uma das causas da lentidão do Software. Neste ponto, a nova versão deu um enorme passo sobre a versão anterior.

Em relação ao Windows 386, este necessita de um Microprocessador Intel 80386 o compatível com 1Mb (embora seja de preferir 2Mb) de RAM, incluindo memória extendida ou espândida, uma drive de 1.2Mb e de 2Mb de disco e corre sobre versões 3.1 do DOS ou versões posteriores.

INSTALAÇÃO

Nesta versão, a instalação apresenta-se bastante mais simplificada pelo facto de se pôr à disposição do utilizador menus de escolha fácil sendo as especificações acerca do material que possivelmente se poderá usar bastante mais conclusas.

É também importante salientar que poderemos instalar esta nova versão sobre uma disquete de 5.25" com 1.2Mb, ou sobre duas disquetes com 720Kb ou ainda sobre duas disquetes de 360Kb, embora neste último caso o módulo de spooler e os ficheiros font não sejam instalados.

No caso de se possuir um disco rígido (o caso em que qualquer das versões do Windows apresenta melhor performance) é de notar que não são necessárias alterações, ou pelo menos, grandes alterações, aos ficheiros AUTOEXEC.BAT e CONFIG.SYS. É possível ainda colocar a chamada ao Windows no AUTOEXEC ou mesmo instalá-lo sobre um segundo disco rígido sem que isso afecte o sistema. O mesmo já não se poderá dizer do Windows 386 que exige a sua colocação no disco principal do sistema.

Em relação aos numerosos parâmetros necessários ao Windows (tais como selecção de programas a executar. selecção de impressoras, portas de Input/Output, etc.), é ainda de referir que este coloca-os num ficheiro designado WIN.INI que qualquer utilizador poderá editar com um vulgar processador de texto para os alterar. Se por outro lado, o utilizador tiver medo de deteriorar o ficheiro, poderá ainda alterar a configuração através do painel de controlo que pode ser acedido chamando o programa CONTROL.EXE.

NOVAS CARACTERÍSTICAS — PERFORMANCES

Ao falar da performance de uma determinada aplicação é importante dizer que a capacidade de processamento de um computador não é ilimitada. À medida que aumenta o número de programas a funcionarem concorrentemente, baixa a quantidade de tempo de CPU para cada aplicação e a execução torna-se cada vez mais lenta. Aliás, este é um dos pontos a chamar a atenção no windows. Na sua primeira versão, a sua lentidão era demasiado evidente. Nesta última versão, 2.03, as coisas já são um pouco diferentes, e, no que toca ao Windows 386, já pouco teremos a dizer. Contudo, ao compararmos os tempos de execução do DOS com o Windows 2.03 e com o Windows 386 podemos verificar que as duas primeiras executam os programa mais ou menos à mesma velocidade, dependendo da memória disponível, enquanto que o Windows 386 demora um pouco mais à medida que vamos tendo mais aplicações a correrem "simultaneamente", em relação ao tempo que o DOS demoraria a executá-las.

Notória é também a maior facilidade de uso do teclado para quem não dispõe de um rato. Nesta nova versão basta carregar na tecla ALT para que se chegue ao menu, enquanto que na versão anterior era necessário carregar na tecla ALT e na primeira letra do menu que se pretendia chamar.

É também de pôr em causa a compatibilidade do Windows com programas residentes tipo SideKick. Em relação a este último, podemos verificar que também não faz muita falta, visto que também temos no windows uma calculadora, um calendário ou ainda um NotePad apesar da sua chamada implicar mais tempo de carregamento. É claro que nem sempre aquilo que temos à disposição no Windows chega para substituir um programa residente, mas é possível, com o Windows 386 e com ficheiros PIF chamar ficheiros Batch que por sua vez invocam aplicações e alguns utilitários residentes. Ao sairmos da aplicação e ao fecharmos a janela, o

Windows 386 encarrega-se de remover tudo o que foi carregado com a aplicação libertando assim a memória ocupada e libertando-se de possíveis incompatibilidades.

COMPATIBILIDADE ENTRE VERSÕES

No programa Paint, os ficheiros da versão 2.03 não são compatíveis com os da versão anterior, não sendo possível passar directamente ficheiros desta nova versão para a anterior. É possível fazê-lo se usarmos o Cut e o Past ou ainda o programa CVTPAINT. EXE (abreviação de Convert Paint). É de notar que, no entanto, é possível ler ficheiros da versão antiga nesta nova versão.

Entre o Windows 2.03 e o Windows 386 não tenho conhecimento de algumas incompatibilidades. É possível passar ficheiros de um para o outro sem que isso ofereça algum problema, e é ainda possível ler ficheiros ASCII puros no Write em qualquer das versões.

O WINDOWS COMO AMBIENTE DE PROGRAMAÇÃO

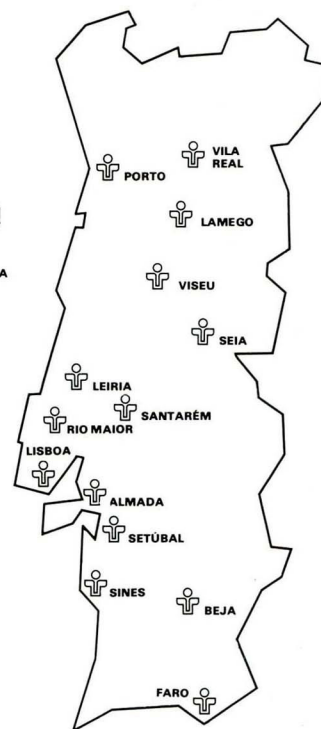
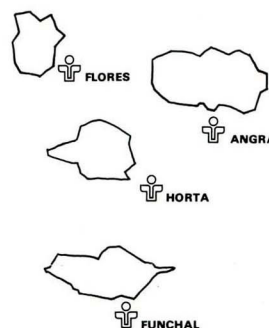
Talvez por deformação profissional, considero a abertura do Windows ao desenvolvimento de novas aplicações que sobre ele funcionem, de enorme interesse. E isto porque para o programador, o Windows providência uma tremenda oportunidade para aprender o funcionamento interno de um dos programas mais bem sucedidos alguma vez desenvolvidos.

O Kit de desenvolvimento da primeira versão do Windows requeria a seguinte configuração: um compatível IBM XT ou AT, uma drive configurada como A, um disco rígido C, 512Kb de RAM (embora recomendados sejam os 640Kb), o DOS 2.0 ou versão mais recente e o compilador de C, Pascal ou o Macro Assembler (MASM) da Microsoft. Embora estas linguagens limitem um pouco as áreas de desenvolvimento (pois nem toda a gente sabe Pascal ou C e muito menos Assembler), é natural que o novo toolkit para abertura para desenvolvimento, embora ainda não esteja disponível, traga já uma abertura para outras linguagens. Isto poria à disposição de mais alguns programadores a possibilidade de desenvolver programas mais "profissionais" (ou pelo menos, com aparência disso), o que poderia tornar a comercialização do Windows ainda mais vulgar.

CONCLUSÕES

Numerosas companhias têm reconhecido a importância do interface gráfico do Windows e a sua popularidade crescente sugere uma importante regra no futuro do desenvolvimento de sistemas operativos e aplicações. No entanto tenho certa dificuldade em entender a política da MicroSoft ao lançar com poucos meses de intervalo a nova versão do Windows, o DOS 4.0 e anunciar ainda o Presentation Manager do OS/2, afirmando ainda que este se parece em muito com o Windows 2.03.

De qualquer maneira, a nova versão do Windows é bastante mais rápida do que a versão anterior e faz o melhor uso de computadores que não possuam o 80386. a versão para o 80386 oferece uma verdadeira multitarefa graças ao poder destes chips e está apta a fazer frente ao OS/2 em muitas frentes, uma vez que este é demasiado exigente em termos de necessidades de Hardware. Esperemos para ver o que acontece...

Paulo Pinheiro

A ATRIBUIÇÃO DE MEMÓRIA NO MS-DOS

Esta função pode ter uma utilização prática num programa TSR inteligente que se se pode 'desinstalar' a si próprio de qualquer parte da memória.

UMA das funções mais importantes dum sistema operativo é a de controlar a utilização da memória pelas várias aplicações.

Basicamente, o MS-DOS constroi sózinho uma lista encadeada de blocos de memória não atribuída e, sempre que é pedida mais memória, o MS-DOS verifica a lista até encontrar o próximo bloco disponível para atribuição. A partir da versão 3.0 passou, no entanto, a haver um método de controlo da estratégia de atribuição de memória, que é um procedimento utilizado pelo MS-DOS para determinar onde irá atribuir, nesta cadeia, um novo bloco de memória. As três estratégias disponíveis para o MS-DOS são:

"o primeiro encaixe", "o melhor encaixe" e "o último encaixe".

Na estratégia do *"primeiro encaixe"*, o DOS pesquisa a cadeia de blocos de memória, desde o endereço mais baixo até ao mais elevado, atribuindo a memória pedida ao primeiro bloco de memória com tamanho suficiente para satisfazer o novo pedido. Esta era a única estratégia disponível para o MS-DOS, das versões da série 2 e é também a estabelecida, inicialmente, pelo MS-DOS 3.x quando o sistema é inicializado. Sob a estratégia do *"melhor encaixe"*, o MS-DOS pesquisa todos os blocos de memória e procede à atribuição no bloco mais pequeno, mas que tenha tamanho suficiente para satisfazer o pedido efectuado pela aplicação. Quando a estratégia *"último encaixe"* é utilizada o MS-DOS pesquisa a cadeia de blocos de memória desde o endereço mais elevado até ao mais baixo e atribui a memória no bloco que esteja mais elevado (na cadeia) e que seja suficientemente grande para corresponder ás necessidades de memória. Se este bloco de memória for superior (em tamanho) ao que foi pedido, a memória é atribuída a partir dos endereços mais altos do bloco; nos dois outros casos, pelo contrário, a atribuição de memória começa sempre a partir do endereço mais baixo.

Esta estratégia de atribuição de memória pode ser verificada ou alterada através da função 58H da interrupção 21. Além disso, está documentada pela Microsoft no 'MS-DOS Programmer's Guide' (Guia do Programador do MS-DOS), mas não é citada pela IBM no seu manual "DOS Technical References". Falando ainda de referências, é interessante ver que é esta a única função constante do manual IBM que é rotulada de 'Used internally by DOS' (Utilizada internamente pelo DOS); todas as outras funções não documentadas são rotuladas de 'Reserved for DOS' (Reservadas para o DOS).

A sequência de chamada para esta função é a seguinte:

MOV AH,58H
MOV AL,00H; para obter a estratégia de atribuição
ou
MOV AH,58H
MOV AL,01H; para a fixar

neste último caso **BL** será igual a **00H** para *"o primeiro encaixe"*, igual a **01H** para *"o melhor encaixe"*, e igual a **02H** para *"o último encaixe"*.

O código de estratégia a seguir é retirado de **BL**, e não de **BX** como é indicado nos manuais da Microsoft e em muitos outros livros. Entrando na estratégia de atribuição (**AL=00H**), o valor de entrada em **BL** é ignorado, e o sistema devolve **0000H**, **0001H** ou **0002H** no registo **AX**, indicando assim a estratégia de atribuição de memória activa.

Se durante a execução da rotina for detectado qualquer tipo de erro a flag de carry é colocada a 1 no retorno, dando-nos assim a hipótese de controlar o sucesso da operação. O único erro que é devolvido por esta função é **AX=0001H**, *Função Inválida*, se o valor de entrada em **AL** não for **00H** ou **01H**. O valor em **BL** não é verificado e, qualquer valor maior que **01H** tem o mesmo efeito que **02H**. Contudo, a próxima chamada para verificar a estratégia de atribuição de memória devolve o último código estabelecido e não o valor efectivo. Assim, um programa que verifique a estratégia de atribuição de memória não deve testar o valor específico de 2, mas sim qualquer valor superior a 1, para determinar se a estratégia *"último encaixe"* está a ser implementada ou não.

A diferença entre as estratégias *"primeiro encaixe"* e *"melhor encaixe"* só é significativa se existirem diversos blo-

cos de memória livres. O MS-DOS reúne sempre blocos livres adjacentes num só, de maneira que só existem diversos blocos livres se não forem adjacentes uns aos outros.

Proceder ao 'desalojamento' de um programa residente, como é sugerido pelo manual da IBM "DOS Technical Reference", através da interrupção 27H e da função 31H, pode deixar na memória os referidos 'buracos' de espaço não atribuido entre os diversos blocos de memória reservada. Contudo, é mais frequente ocorrer um espaço de memória fragmentada num sistema multitarefa (como o Prologue ou o Xenix). Só em circunstâncias muito pouco usuais é que isso acontece num sistema de mono-tarefa (como o MS-DOS).

Apesar disto, a estratégia *"último encaixe"* difere, na realidade, do procedimento inicial, mesmo com um único bloco de memória livre, desde que o bloco de memória requerido seja mais pequeno que a memória disponível. Nesse caso, a atribuição é feita a partir do topo do bloco livre. Para ilustrar este efeito, introduza-se e execute-se o programa a seguir apresentado dentro do utilitário DEBUG:

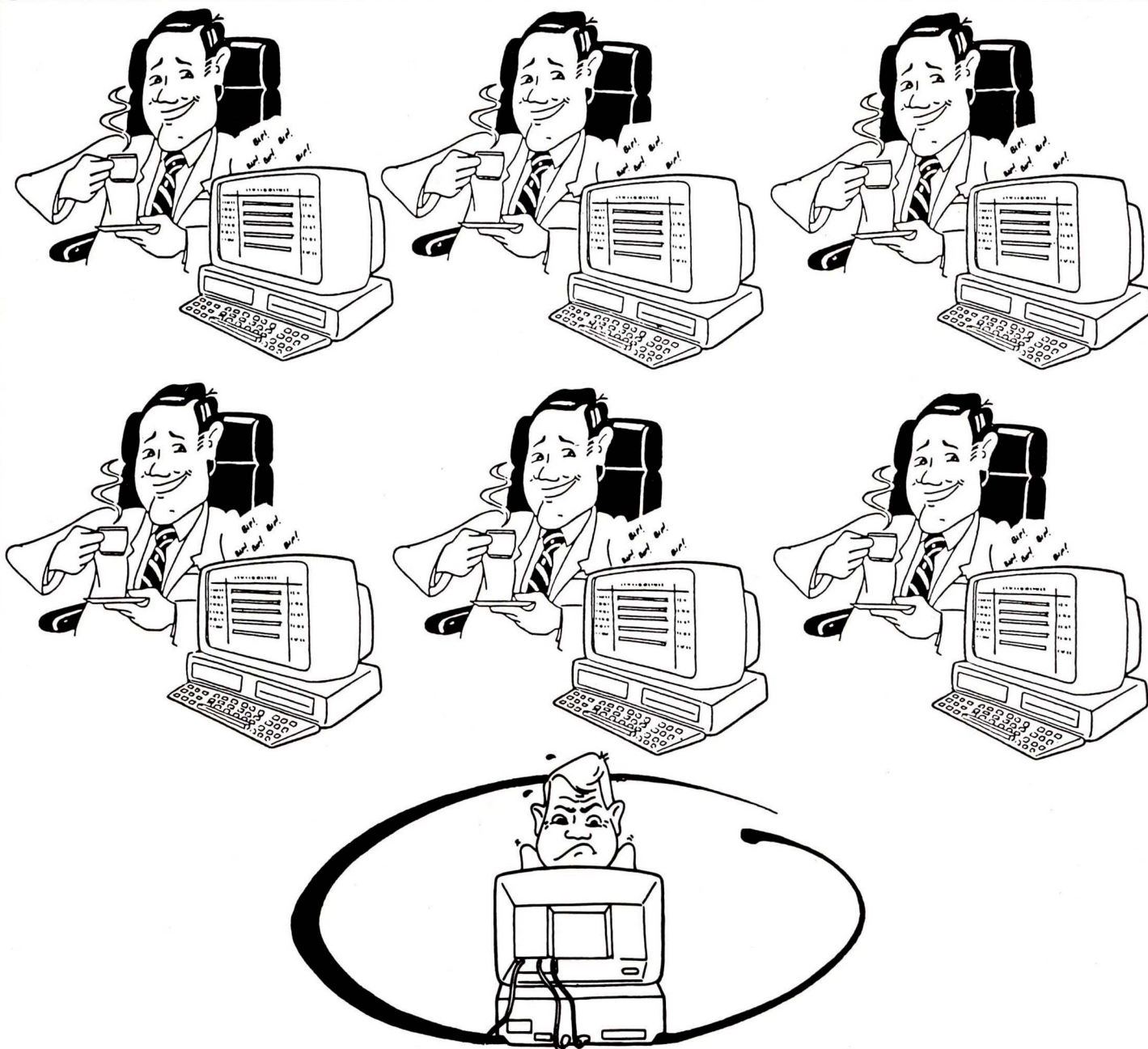
```
MOV AX,5801
MOV BX,2
INT 21
INT 20
```

Agora saia do DEBUG e execute um CHKDSK. O total de memória disponível é inferior ao habitual - por exemplo, num computador com 640 Kb, o total pode ser de 655120 bytes em vez dos habituais 655360 (640 x 1024 bytes). A redução exacta depende da dimensão do bloco destinado a construção do ambiente de trabalho.

Quando o processador de comandos do DOS carrega um programa, reserva dois blocos de memória para esse efeito. Primeiro reserva um bloco de tamanho suficiente para que possa suportar uma cópia do ambiente de trabalho, contendo as cadeias alfanuméricas que especificam os COMSPEC, PATH, PROMPT e quaisquer variáveis criadas por comandos SET. Numa estratégia de atribuição de memória do tipo *"último encaixe"*, este espaço é reservado no topo da memória. Em segundo lugar, o maior bloco livre é atribuído ao próprio programa. O CHKDSK indica o total de memória disponível como sendo o fim da memória em que está a executar-se, baseado na pressuposição de que o maior bloco livre se estende até ao topo da memória.

Uma vez que o carregador de programas atribui sempre todo o maior bloco livre a um programa, a estratégia de atribuição não tem geralmente qualquer efeito na localização de um programa em memória, excepto quando existam diversos blocos com a mesma dimensão máxima; nestes casos, o programa é então carregado no bloco de memória com os endereços mais altos, sob a estratégia *"último encaixe"*, sendo, no entanto, em todos os outros casos, colocado no bloco com endereços mais baixos.

Uma utilização prática para esta função poderá ser um programa TSR (Terminate and Stay Resident) inteligente, que se pode desinstalar a si próprio de qualquer parte da memória, mesmo que não tenha sido o último programa a ser carregado. Quando houver uma re-instalação posterior, este programa poderá desencadear uma estratégia de atribuição de memória do tipo *"o melhor encaixe"* de modo a localizar um espaço de memória ideal e re-colocar aí a sua porção residente, evitando que esta vá parar onde quer que o DOS a tenha carregado.

Você não quis perder um minuto !

Decidiu comprar software sem determinar primeiro
as necessidades da sua empresa.
JÁ MUITOS NOS CONSULTARAM
Mas o seu caso não é só mais um caso para nós.
Por isso, concebemos programas adequados
à sua situação específica.
PERDER UM MINUTO É GANHAR... MUITO TEMPO!

A solução

Shopping Cacém, loja 2.42 Telef. 928 09 29 Cacém

mahe

Em Busca da EXCELência

Toda a gente conhece a marca Singer. Mas já nem toda a gente sabe o que é a European Home Products. No entanto, os dois nomes estão intimamente relacionados, para já através de uma extensa cadeia de lojas espalhadas pelo país. No ano passado apareceu então a EXCEL, uma nova cadeia do grupo EHP, dedicada á venda de diversos tipos de electrodomésticos, incluindo audio e video. E, confirmando as previsões dos analistas, o computador pessoal acabou também por entrar no negócio, tipo banalização da informática. Uma das marcas que a EXCEL comercializa é a Amstrad, nomeadamente os já conhecidos 1512 e 1640 e o novo 2086.
A Amstrad Magazine foi então conversar com dois importantes executivos da EHP e da EXCEL: António Costa, gerente do grupo, e Mário de Oliveira, director de marketing e produtos da EXCEL. O assunto? É claro, dar a conhecer melhor a empresa ao público leitor da AM e saber como é que se consegue vender frigorificos e computadores ao mesmo tempo.

António Costa (à direita): "vamos abrir brevemente lojas mais dedicadas á informática e com novos produtos e marcas".

Como e quando é que apareceu a EXCEL? E com que objectivos?

António Costa: A EXCEL apareceu como necessidade de preencher um espaço que se encontrava em aberto no ramo de logistas (retalho) de electrodomésticos em Portugal. Após algum trabalho de pesquisa de mercado e de avaliação do projecto em termos de desenvolvimento passou-se de imediato à implementação do mesmo.

Em Abril de 1988 foi aberta ao público a primeira loja no Centro Comercial das Amoreiras e passados que foram 11 meses já temos em funcionamento nove lojas - oito na área de Lisboa e uma no centro do Porto. É nossa intenção fechar o ano de 1989 com 20 lojas em funcionamento em todo o país, sendo tudo isto possível pelo grupo de trabalho que está à frente das operações da EXCEL, e que a este projecto tem dedicado praticamente toda a sua vida pessoal e profissional.

Hoje em dia verificamos que o mercado dos electrodomésticos vive um clima de instabilidade, decorrente das formas destorcidas como grande parte dos retalhistas funciona em Portugal, não dando ao cliente a satisfação de algumas das suas mais elementares exigências; como por exemplo, proporcionar-lhe um conjunto de serviços que não se resumam ou esgotem com o acto da venda. Ora, o nosso principal objectivo é exactamente preencher essa lacuna.

Que tipo de produtos é que a EXCEL vende?

Mário de Oliveira: Nós comercializamos uma gama de produtos bastante diversificada, ao nível da linha branca (fogões, máquinas de lavar, frigorificos, etc), castanha (YV, Video e Hi-Fi), encastráveis, pequenos electrodomésticos, instrumentos musicais e equipamento informático.

É importante salientar que em todas as nossas lojas, cada uma destas linhas de produtos se encontra distinta de cada uma das outras através de um tratamento de 'merchandising' adequado à sua especificidade.

Pode pormenorizar um pouco melhor esse aspecto?

A.C: O 'merchandising' em Portugal ainda não tem sido, como devemos compreender, devidamente tratado. Não quero dizer com isto que não existam já algumas lojas especializadas que se aproximem ou estejam mesmo num estádio que podemos considerar perfeito. 'Merchandising' tem um sentido muito lato. Pode passar por ponto de venda, pode passar pela disposição do produto, pode passar pela apresentação da loja, pela própria limpeza, etc. Quando nós entendemos que o merchandising tem que ser melhorado, sentimos que em Portugal não existe esse tipo de operação, esse tipo de planeamento que é pôr o produto da forma correcta. Para isso há regras fundamentais. Não vou, logicamente, descrevê-las porque isso faz parte dos manuais. Direi apenas que na nossa companhia temos 4 'merchandisings'. E bem sei que a companhia é larga e envolve não só a reparação e beneficiação dos pontos de venda [lojas], mas também a concepção do material de exposição e bem assim qual o melhor local para pôr o produto A ou o produto B, qual é a melhor forma para sensibilizar o consumidor ao produto de baixa gama, de média gama ou de alta gama... Tudo isto pode ser feito em termos de ponto de venda. E não tem sido feito - ou se põe o produto porque é aquele que neste momento está em chamariz e se coloca duma forma frontal na loja, ou mistura-se o hardware e o

software duma forma confusa nas montras etc.

Há pouco falava de lojas especializadas para informática. A EXCEL planeia seguir por esse caminho?

A.C: Vamos fazê-lo. Gostariamos de o fazer já em 1989. E vamos contar naturalmente com a ajuda e com o suporte, que consideramos indispensável, dos fabricantes.

...novas lojas ou adaptando outras que já existam?

A.C: Aproveitaremos as duas situações. Por um lado adaptaremos as lojas que entendemos não estarem a dar o rendimento necessário, ou que desvirtuem o fim a que se destinarão, no campo da informática. E, por outro, lojas novas.

Isto porque o consumidor deste tipo de produtos [informática] é muito especial, um dos mais avançados em termos de exigências. Logicamente, não se gosta de ser misturado e confundido com perguntas do estilo "mas como é que se abre esta porta do frigorifico?"

Que tipo de produtos é que vendem mais?

M.O: A actividade da EXCEL tem um tempo de duração relativamente curto - menos de um ano - e, como tal, as análises que poderíamos fazer neste momento não poderão ser conclusivas. No entanto, face aos resultados já obtidos, poderei afirmar que na linha castanha [TV, video e Hi-Fi] se tem verificado um maior potencial de vendas.

Como é que vêm o mercado actual, nomeadamente no que respeita à concorrência?

A.C: O mercado dos electrodomésticos apresenta tendências para continuar a crescer, embora de uma forma menos acentuada que nos últimos anos. Por outro lado, o mercado dos equipamentos informáticos vai-se desenvolver extraordinariamente nos próximos anos. Neste contexto de aumento da procura, deparamos com uma situação de certa forma incaracterística, que é a da existência de determinados sectores de retalhistas que procuram atrair o consumidor através do único factor que é o preço, esquecendo por completo aspectos essenciais que são exigíveis pelo consumidor e criando, ao mesmo tempo, uma situação de guerra de preços e, consequentemente, uma instabilidade no mercado.

Esta atitude é devida em parte à forma passiva como alguns fabricantes ou distribuidores nacionais de marcas com prestígio assistem a esta situação, não introduzindo eles próprios medidas de correcção tendentes a que os produtos sejam tratados duma forma profissional e assim o consumidor possa receber o serviço que lhe é devido.

O que é que quer dizer com isso?

A.C: Ora vamos lá ver. Duma forma geral nós procurámos posicioanarmo-nos primeiramente na globalidade do mercado, incluindo naturalmente as linhas que comercializamos, ou seja, não especificamente a área de informática. É evidente que o mercado da informática vai evoluir: novas lojas vão existir e quando nós dizemos que haveria áreas ou situações especificas, entendemos que muito há ainda a fazer dentro do aspecto da informática de melhorar a qualidade de atendimento e serviço que tem que se dar ao consumidor final. Porquê? Porque entendemos que existe mais do que vender o hardware, é possibilitar mais do que isso. Pode fazer a pergunta - Estamos nós [EXCEL] a fazê-lo hoje? - Não estamos! Estamos ainda a uma grande distância daquilo que entendemos tem que ser feito. Pelo contrário, não estamos a fazer nada comparado com aquilo que são os nossos objectivos neste campo. Sentimos, contudo, é que o mercado tem que evoluir nesse sentido. O consumidor tem que ter serviço. E o serviço passa naturalmente por um atendimento eficiente, por pessoal experiente. E a formação é aqui fundamental, porque sem ela não vamos lá.

Mário de Oliveira: "Os consumidores de informática são exigentes porque são, normalmente, profundos conhecedores".

Formação interna ou a partir dos fabricantes?

A.C: Logicamente, esperamos que os fabricantes se envolvam. E têm que se envolver. O retalhista, com a força que irá ter no mercado a médio prazo no mercado português, irá contribuir para forçar o próprio fabricante a definir-se. Não pode dar só hardware, tem também que oferecer serviços, de formação do pessoal que trabalha nos pontos de venda e deixarmos de ter um certo sentido familiar - que é importante e necessário, mesmo fundamental - mas que se coadune com uma filosofia de negócio duma cadeia e duma rede de vendas.

Quais são os vossos maiores problemas neste momento?

A.C: A resposta a essa pergunta prende-se com a anterior, pois o maior problema é de facto conseguir impor uma nova forma de estar no mercado, que ainda possui alguns vícios. De qualquer forma, o consumidor já se vai apercebendo da situação e reagindo de uma forma bastante positiva em relação a nós, sendo previsível que em 1989 a situação se vá clarificando e melhorando. Isto porque já vão aparecendo novos agentes a trabalhar duma forma profissional e por outro lado os representantes nacionais das principais marcas terão forçosamente que apostar mais claramente naqueles que tra-

Loja da EXCEL nas Amoreiras. Ao observador desatento poderá passar despercebida a cuidada arrumação e posicionamento dos produtos. Contudo, os executivos da empresa ainda não estão satisfeitos e pretendem que o 'merchandising' seja ainda melhorado.

balham os seus produtos duma forma honesta e competente.

O que é que representam os produtos informáticos na vossa empresa, em termos de vendas e volumes de negócios?

A.C: Neste momento é prematuro fazer uma análise correcta da representatividade das vendas dos produtos informáticos no volume global de vendas, atendendo a que só em Julho de 1988 introduzimos um modelo (O PC1 da Olivetti) e em Novembro arrancámos com a linha da Amstrad.

De qualquer forma e face aos dados disponíveis neste momento, não tenho dúvidas em afirmar que no presente ano os produtos informáticos representarão uma das linhas de produtos mais importantes no nosso volume global de vendas, até porque pensamos alargar o número de modelos e marcas, a comercializar em tempo muito próximo.

Acham que é diferente vender computadores e periféricos e vender outro tipo de produtos?

M.O: Não é substancialmente diferente de alguns outros produtos, nomeadamente aqueles que têm requisitos técnicos e formas de funcionamento electrónico com alguma sofisticação. A diferença mais notória é devida ao facto de o computador, ao contrário de outros produtos, poder servir para diversos fins, o que requer da parte do vendedor um conhecimento profundo das potencialidades do equipamento em cada um dos fins possíveis - para mais sabendo-se que ele é dirigido a um tipo de consumidor exigente porque, normalmente, profundo conhecedor.

Neste aspecto, costumam fazer uma formação especial para os vendedores dos produtos informáticos?

M.O: A formação profissional dos nossos vendedores merece a maior atenção da nossa parte. Sobre isso poderei informar que todos os elementos da equipa de vendas são sujeitos a um periodo de formação junto dos nossos fornecedores, para aquisição de conhecimentos relativos aos produtos, para além da formação interna, onde são devidamente preparados para prestarem um serviço qualificado nos pontos de venda.

Relativamente aos produtos informáticos, as regras mantêm-se, havendo a considerar que destacamos de cada loja um ou dois elementos que receberão depois formação mais específica na área doa produtos informáticos.

Como é que fazem a assistência dos produtos informáticos que vendem?

M.O: A assistência aos produtos informáticos que vendemos tem para nós dois aspectos a considerar: encaminhamento e acompanhamento. Isto porque não limitamos a nossa acção à indicação ao cliente aonde se deverá dirigir para a reparação ou assistência em garantia dos produtos, e que é, normalmente, o departamento de assistência técnica das marcas. Além disto, procuramos ainda estar em contacto directo com esses serviços, de modo a que o problema do nosso cliente seja solucionado no mais curto espaço de tempo possível.

Prevêm um maior crescimento nas vendas e negócios da área de informática ou noutras áreas (video, audio, electrodomésticos, etc)?

A.C: Creio que o ano de 1989 não será um ano particularmente favorável ao aumento do consumo. Contudo, as áreas de mercado que trabalhamos apresentarão níveis de procura que nos podem permitir um nível de crescimento razoável em todas as linhas que comercializamos.

Reconheço, no entanto, que é exactamente no equipamento informático que esse crescimento será mais evidente, pois é notória uma, cada vez maior, procura destes produtos, quer ao nível profissional, quer ao nível particular.

QUEM TRAMOU ROGER RABBIT

Depois do sucesso mundialmente atingido com o filme, eis que surge o jogo para diversos modelos de computadores, transportando para o mundo dos digitos uma eterna "gargalhada analógica".

HOLLYWOOD, 1947, uma história de "desenhos animados" - ou "cartoons" (como preferirem) -, algumas pessoas, muita imaginação, e nunca menos loucura. Este é, mais ou menos, o ambiente escolhido para o simpático Roger se tornar o herói da miudagem. O filme que continua a passar em alguns cinemas portugueses parece ter começado a sua carreira apenas ontem e continua a manter plenas de calor humano as salas por onde passa. O jogo vai mesmo começar agora o seu percurso e, garantimos desde já, não vai envergonhar de modo algum o coelho que conhecemos no grande ecrã.

A HISTÓRIA DE UM COELHO CHAMADO ROGER

Para os leitores que não tiveram oportunidade de ver o filme, podemos fazer em duas ou três linhas um "retrato" da história de "Quem tramou o Roger Rabbit".

Assim, Roger é um coelho (daí o seu nome de familia "Rabbit") que trabalha para a empresa cinematográfica Maroon Cartoon Studios (Estudios de Animação Maroon). É claro que Roger não se limita a trabalhar para a Maroon Studios, Roger é a estrela da Maroon Studios, e como tal toda a acção se desenrola em seu redor. À noite, terminadas as filmagens, estudados todos os pequenos pormenores das cenas e dos roteiros, Roger regressa a casa, nos divertidos suburbios de Toontown ("Cidade da Animação"), onde as piadas e as brincadeiras parecem nunca ter fim. Tudo parece funcionar às mil maravilhas até ao dia em que alguém decide tramar Roger, incriminando-o como o assassino de Marvin, o rei dos "gags" e proprietário de Toontown, que entretanto tinha aparecido misteriosamente morto. Roger sentindo-se em perigo foge, mas o assunto torna-se mais grave do que à partida parecia pois apesar da morte de Marvin poder vir a afectar "as boas condições de saude de Roger", o ex-proprietário de Toontown tinha prometido deixar a sua cidade aos "cartoons" que nela viviam, num testamento que nunca alguém tinha visto, mas que tudo levava a crer que existia. Resumindo, com Marvin morto, Roger em fuga por ser procurado pela morte de Marvin, e o testamento de Marvin desaparecido, chegavam os dias negros para todos os "cartoons". A juntar a todos estes azares, o juiz Doom, com o seu pretexto de "administrar justiça" nunca perdia uma boa oportunidade de eliminar mais "cartoons" mergulhando-os numa mistura quimica letal, a que decidiu chamar "o banho" ("the dip"), que corrói imediata e totalmente os "cartoons" que nela tocam. Para Doom o prazer máximo seria destruir Toontown, se possivel eliminando todos os "cartoons". Para Roger o prazer máximo seria destruir Doom deixando Toontown intacta, se possivel sem sacrificar a sua "deliciosa" mulher Jessica, nem cometer "suicidio involuntário".

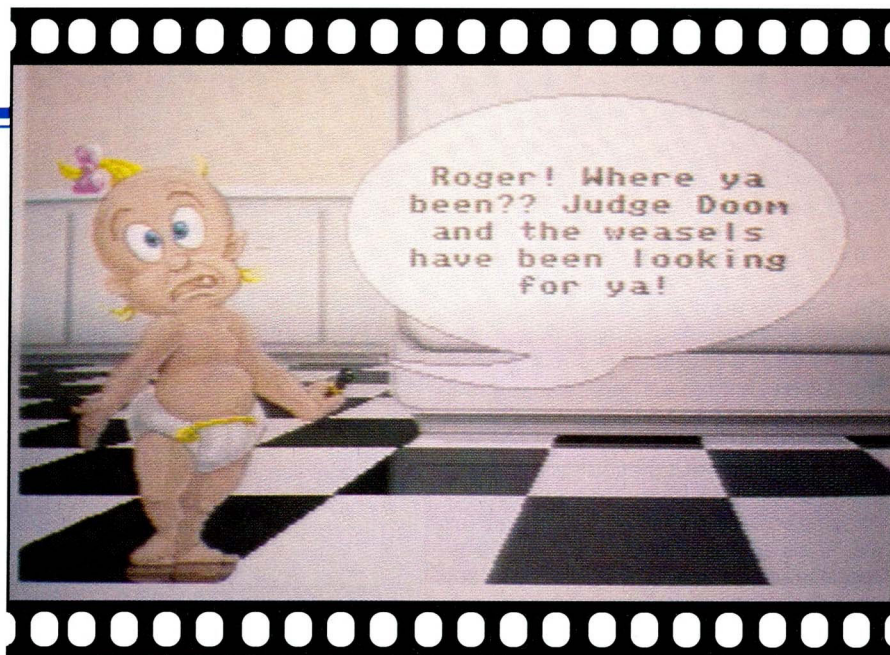
Nesta luta, quase transformada numa guerra pessoal entre Doom e Roger onde vale tudo menos tirar olhos, cabe ao leitor/jogador ajudar uma ou

outra personagem a vencer, salvando Toontown, ou contribuindo para retirar esta cidade do mapa. Como é lógico o objectivo do jogo é salvar Toontown, ajudando a ilibar Roger do crime cometido por alguém que à partida desconhecemos, encontrando o testamento de Marvin, e evitando a morte de Jessica (se assim não fosse, o titulo do jogo seria, muito provavelmente, "Ajudem a tramar Roger Rabbit").

Roger Rab**bit** ou Roger Rab**byte**

Com diversas versões já disponiveis para computadores tão diferentes como o Atari, o Commodore Amiga, e o PC compativel IBM, por exemplo, o jogo em causa chegou-nos em formato 3.5" e 5.25", respectivamente, para as duas últimas máquinas aqui referidas. Sem qualquer semelhança possivel em termos de capacidades gráficas e sonoras, o Commodore Amiga e o PC compativel, possuem talvez as versões extremas no aspecto de qualidade final do jogo analisado. Para os menos conhecedores do Amiga, podemos informar que contrapondo às 4 cores simultâneamente disponíveis no ecrã dos PC´s em modo gráfico CGA, o Amiga pode dispor de 4096 cores simultâneas, e que para competir com o único canal sonoro dos PC´s o referido computador pode utilizar 4 canais de som estéreofonicos. Diferenças "insignificantes" como podem observar pelas imagens que apresentamos, e como poderão constatar brevemente nas lojas onde costumam adquirir o vosso software. Ainda assim, apesar destas diferenças, a versão PC do jogo em causa encontra-se extraordinariamente bem conseguida sob o aspecto gráfico, levando-nos mesmo a afirmar sem receio que: se todos os jogos (e mesmo aplicações) em CGA fossem tão cuidados sob o ponto de vista gráfico como o "Roger", o número de utilizadores de EGA´s e VGA´s seria consideravelmente inferior.

HaaaaAAAAH isto é que é CGA!!!?

Na realidade a carta CGA parece muito melhor quando o pequeno coelho assume o seu controlo. Dos muitos programas que temos visto para esta carta gráfica nada de tão bem conseguido graficamente se nos apresentou até ao momento. Este aspecto é, aliás, o ponto mais forte do jogo, que leva mesmo a soft house que o concebeu a designá-lo, não por um simples jogo mas, por "entertainement software" (software para divertimento). O jogo "Who framed Roger Rabbit" é, portanto, para além do jogo um "filme", uma história, enfim, acima de tudo um divertimento. A "qualidade" do divertimento varia nitidamente com as capacidades da máquina em que este se encontra a "correr". Como é lógico, os minutos passados em frente do Amiga, são muito mais divertidos do que os que passamos em frente de qualquer PC, mas também não podemos esperar que se façam milagres. Já podemos considerar excelente, tudo o que foi conseguido com apenas quatro cores e uma resolução de 600x200. Nesta área não podemos esquecer que a Buena Vista Software é uma empresa do grupo Walt Disney, e que como tal não deixa o crédito por mãos alheias em termos de aproveitamento das capacidades gráficas dos computadores. Outros aspectos do jogo foram menos bem conseguidos, mas nunca o aspecto gráfico, que não nos cansamos de repetir faz destas duas disquetes de 5.25" um "must" para todos os possuidores de cartas gráficas CGA, e um jogo muito apetecivel para todos os utilizadores que dispõem de uma carta gráfica com mais capacidades (EGA, VGA, etc.).

Como consequência principal desta grande qualidade gráfica, o Roger em PC perde irremediavelmente velocidade. Apesar de tudo, a velocidade atingida num PC, embora seja inferior à que nos espera num Commodore Amiga, é suficiente para nos proporcionar um jogo-espectáculo muito bom, sem deixar que o jogo caia no dominio dos jogos impossiveis. Por outras palavras passamos a explicar melhor esta ideia. Alguns dos jogos do mercado são demasiadamente dificeis de concluir, e obrigam os jogadores a travar dificeis lutas com o teclado afim de procurar "ver mais qualquer coisa". Este não é um desses jogos. A título de curiosidade, diga-se de passagem que com um minimo de habilidade, algum tempo de dedicação, e um pouco de "batota" conseguimos ver todos os ecrãs do jogo ao fim de três dias de "trabalho". A facilidade de concluir o jogo, por sua vez aliada ao requinte gráfico que caracteriza cada ecrã, aos comentários do Herman, da Jessica, ou mesmo do Roger, e à musica diferente para cada nível de jogo, proporcionam a cada momento um espectáculo que nunca antes tinhamos observado em CGA.

O JOGO PASSO A PASSO

Depois de inicializarmos o jogo, de termos seleccionado o monitor RGB e o teclado, ou o joystick, como forma controlar o nosso Roger, e depois vermos todos os ecrãs dedicados à apresentação do jogo (se desejarmos "saltar" estes ecrãs basta premir a barra de espaços), entramos finalmente no primeiro nível de jogo.

Benny - o táxi

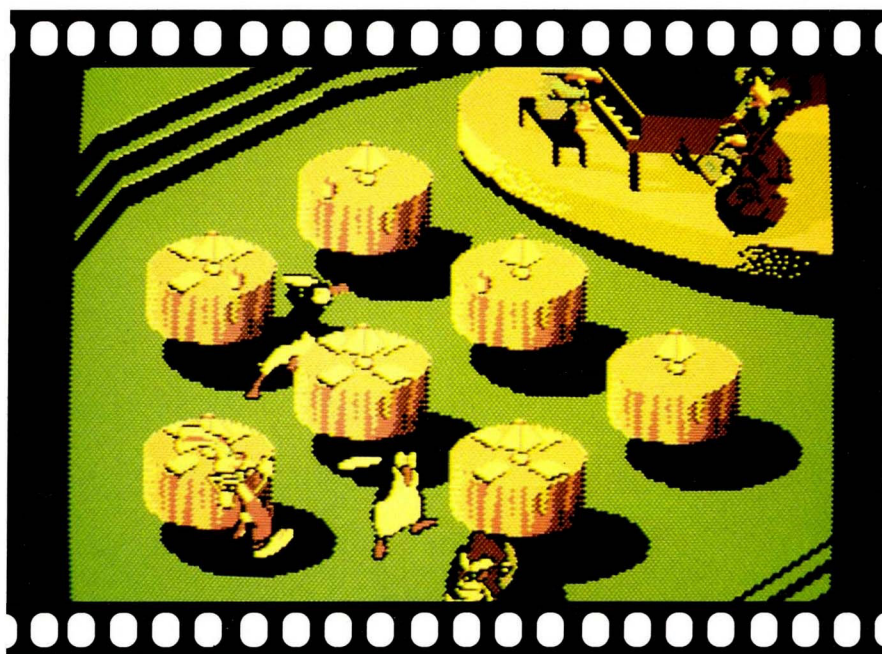
Roger, acompanhado pelo dedicado amigo Benny (um táxi de Brooklyn com caracteristicas muito especiais), percorre as ruas de Hollywood, tentando chegar primeiro do que o juiz Doom ao clube "Ink and Paint" ("Tinta & Pincel"). Na desenfreada corrida contra o tempo e contra todos, tanto Roger como Benny têm de ter muito cuidado com as poças de "banho" espalhadas ao longo do percurso, com as viaturas conduzidas pelos fuinhas ("tipos" com um ar sinistro, amigos do juiz Doom, e que nós conhecemos desde há muito dos livros Disney), e com as viaturas vermelhas. Qualquer um dos referidos obstáculos pode, no entanto, ser evitado se, usando a fantástica suspensão do Benny saltarmos para o topo dos edificios que se encontram no passeio em frente, continuando então o passeio saltando de terraço em terraço. Deste modo, embora não se percam "vidas", dificilmente se conseguirá chegar ao clube primeiro do que o juiz Doom, e nunca se conseguirão apanhar as luvas de borracha, as rodas, ou os diamantes que se

encontram suspensos nas paredes das casas por onde passamos. Estes objectos são de grande importância para a perfeita conclusão do jogo, já que podem proporcionar uma protecção temporária resistente ao "banho" (luvas de borracha), um aumento da velocidade do Benny (rodas), ou mesmo mais uma vida para facilitar a nossa árdua tarefa (diamantes). As instruções que acompanham o jogo alertam ainda para o facto de se poderem encontrar outros objectos de grande utilidade que devem de igual modo ser apanhados, embora durante todo o tempo de jogo nunca tenhamos detectado mais objectos do que aqueles que referimos. Ainda neste nível, e seguindo as instruções que previamente tinhamos lido, não conseguimos observar uma reacção que nos é apontada como estranha ao saltarmos para o tejadilho dos carros vermelhos. Talvez com mais algumas horas de "treino intensivo" se consiga comprovar o que as instruções nos apontam.

O clube "Ink and Paint"

Benny deixado à porta a descansar da grande corrida, e eis que começa a grande corrida de Roger. No clube "Ink and Paint", o único local no mundo onde os humanos podem apreciar as actuações dos "cartoons", e admirar a fabulosa Jessica (mulher de Roger), 7 mesas servidas por dois pinguins esperam por Roger para começar toda a acção. O objectivo é apanhar todos os papeis que os pinguins deixarem em cima das mesas afim de conseguir encontrar o testamento de Marvin, antes dos músicos deixarem de tocar. A dificuldade consiste no facto dos pinguins nunca deixarem de servir as mesas deixando nelas sempre alguns papeis e bebidas. O obstáculo principal é Jonas, o gorila, que passeia silenciosamente na sala e "pesca" todos os clientes indesejáveis (especialmente os que possuem duas grandes orelhas, e um rabo tipo pompom). A advertência final neste nível vai para as bebidas que os pinguins deixam em cima das mesas que nunca, mas nunca mesmo, devem ser ingeridas por Roger.

A fábrica de "gags"

Considerado o terceiro nível nas instruções, apesar de entre este nível e o anterior existir ainda um passeio idêntico ao que inicia o jogo, a fábrica de "gags" é o local onde o jogador tem de jogar com mais dedicação. Alguns passos mal dados, alguns momentos perdidos e todo o trabalho tido até ao momento se encontra desperdiçado. Este é também o nível em que o jogador tem de pensar mais em termos de estratégia a seguir, e ainda aquele em que o riso se torna peça fundamental. Com efeito, é neste nível que temos de destruir os fuinhas numa "luta de riso corpo a corpo", isto é fazendo-os rir o mais possivel. Os fuinhas são "cartoons" tal como Roger e Jessica, e a única forma de os eliminar é provocando-lhes um ataque de riso ininterrupto. Sabendo isto torna-se fácil para Roger eliminar todos os fuinhas usando os diversos "gags" que vai encontrando na fábrica. Porém as bombas suspensas no tecto também podem eliminar os fuinhas, e embora esta tarefa não se possa considerar dificil, não nos devemos esquecer que enquanto vamos fazendo palhaçadas o tempo vai passando, e o juiz Doom não hesita em matar a nossa adorável, inigualavel, carinhosa, deliciosa, sexy Jessica (espero que o Roger não leia este texto senão alguém vai ter de criar a história "Quem tramou Fernando Rabbit").

Sobre este nível muito haveria ainda para dizer, mas sobre os outros também há qualquer coisa de escondido que aqui não foi referido, nem será. No fundo, nós apenas pretendemos mostrar-lhe um bom jogo, e não queremos, de modo algum, estragar-lhe todo o prazer de vestir a pele de Roger durante algumas horas, tentando salvar Toontown e eliminar QUEM TRAMOU O ROGER RABIT.

Fernando Prata

Agradecemos em especial a:
Nanci Rahanasto — Buena Vista Software
Pierre Sissmann — Cedic/Nathan
Raachel — Cedic/Nathan

que nos apoiaram incondicionalmente e satisfizeram todos os nossos pedidos, tornando possivel a análise do jogo neste número da AM, quando a versão PC ainda não se encontra disponivel para venda ao público.

UM 464 MAIS GRÁFICO

O CPC-464 possui inegáveis qualidades gráficas numa resolução de 640x200 pixel´s em modo 2. Embora o excelente BASIC que o acompanha inclua um conjunto relativamente completo de instruções gráficas, sempre há uma ou duas instruções úteis que não estão incluídas no dialecto de BASIC do 464. Uma instrução para desenhar circunferências, e uma outra para colorir grandes áreas do ecrã, estão claramente no conjunto das que acabámos de referir.

O programa **Circunferencias** desenha uma circunferência com 100 pixel´s de raio, com o centro no meio do ecrã. Para alterar a dimensão ou posição da circunferência, modifique as variáveis **r** - comprimento do raio - , **x** e **y** - coordenadas do centro.

O segundo programa - **Circunferências concentricas** - desenha duas circunferências concêntricas, uma com 100 pixel´s de raio, e a outra com 50 pixel´s. Neste caso as variáveis utilizadas são **r** e **r2** - comprimento dos raios das circunferências - , **x** e **y** - coordenadas do centro das circunferências.

Com ligeiras alterações ás rotinas até agora apresentadas é também possível desenhar círculos - coloridos - no CPC-464, bastando para tal desenhar várias linhas dentro da circunferência.

O programa **Circulos** desenha um círculo colorido com um raio de 50 pixel´s, no centro do ecrã - ver variáveis **s**, **x** e **y** - desenhando uma linha na metade superior do círculo que é depois repetida na metade inferior.

Depois de ter observado o programa **Circulos**, execute-o de novo e escreva:

INK 1,1

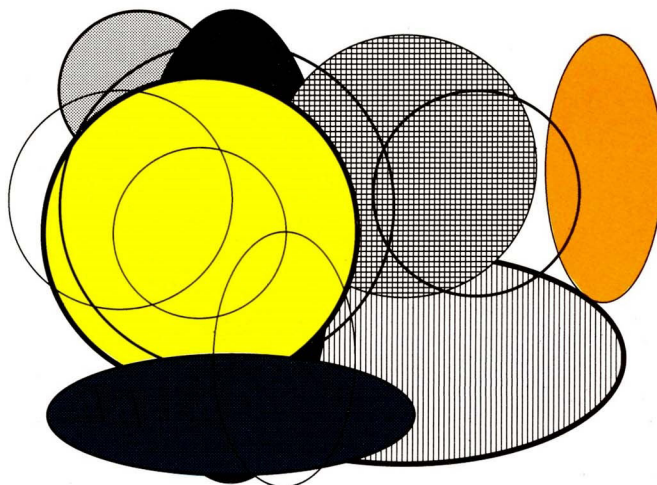
o círculo e o cursor desapareceram! Para os fazer regressar escreva:

INK 1,24

Para brincar mais ainda acrescente as linhas seguintes ao programa **Circulos:**

25 INK 1,2
40 IF s=0 THEN INK 1,24:END
70 IF s=0 THEN INK1,24:END

Quando funciona em modo 1 o CPC-464 dispõe de quatro cores diferentes para trabalhar no ecrã, sendo este o modo de video que ele assume depois de inicializado através do teclado , ou do interruptor de alimentação. O fundo - cor do papel - é inicializado em cor 0, e ao primeiro plano - caneta -, no mesmo momento, é atribuida a cor 1, podendo, no entanto, alterar-se este estado de coisas de acordo com o gosto pessoal.

Nos exemplos por que passámos o círculo deixou de ser visualizado porque passou a ter a mesma cor para os dois planos (papel, e caneta passaram a ser da mesma cor). As

```
10 REM CIRCULOS
20 MODE 1:CLS:CLG
30 R=100:X=320:Y=200
40 DEG
50 ORIGIN X,Y
60 FOR A=1 TO 360
70 PLOT R*COS(A),R*SIN(A)
80 NEXT A
```

```
10 REM CIRCULOS CONCENTRICOS
20 MODE 1:CLS:CLG
30 R=100:R2=50:X=320:Y=200
40 DEG
50 ORIGIN X,Y
60 FOR A=1 TO 360
70 PLOT R*COS(A),R*SIN(A)
80 PLOT R2*COS(A),R2*SIN(A)
90 NEXT A
```

```
10 REM CIRCULOS SOLIDOS
20 MODE 1:CLS:CLG
30 X=320:Y=200:S=50
40 IF S=0 THEN END
50 R=S*S
60 S=S-1
70 IF S=0 THEN END
80 Z=SQR(R-S*S)
90 L=Z+Z
100 SL=X-Z
110 IF SL<0 THEN L=L+SL:SL=0
120 PLOT SL,Y+S:DRAWR L,0
130 IF Y-S<0 GOTO 60
140 PLOT SL,Y-S:DRAWR L,0
150 GOTO 60
```

instruções destinadas a alterar as cores utilizadas no ecrã podem, tal como se deduz pelo que acabámos de dizer (e mostrar), ser usadas de um modo muito fácil para criar efeitos de animação.

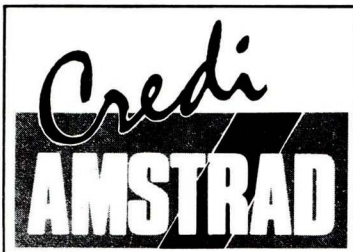
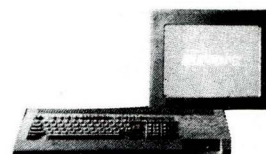
Voltando aos riscos, e deixando as cores por breves momentos, podemos mostrar que também é possível desenhar ovais no CPC-464 apenas com meia dúzia de linhas em BASIC. O programa utilizado é semelhante ao do desenho de circunferências, pois, no fundo, uma elipse não é mais do que uma circunferência alongada.

Na nossa "construção de elipses" a linha oval tem 100 pixel´s de altura, e metade dessa medida de largura (50 pixels). Para alterar esta situação, modifique a variável **r** na linha 40 e o número de **r**'s na linha 70.

O programa **Circunferencias cor** demonstra a utilização do preenchimento de circunferências acompanhado por algumas "brincadeiras" com cores. Utilizámos o modo 0 porque é aquele em que podemos dispor de mais cores (16), embora tenha uma resolução relativamente baixa - 160 X 200 pixel´s.

A tarefa desempenhada pelo programa é facil de descrever. Primeiro este desenha uma circunferência num ponto aleatório do ecrã, com um raio calculado ao acaso. Seguidamente, o bordo do ecrã e as cores utilizadas são mudadas e desenha-se uma nova circunferência. Esta operação é depois repetida várias vezes para criar um efeito decorativo. Para alterar o raio máximo da circunferência, modifique o valor da variável **R** na linha 40. **Circunferencias cor** é um programa unicamente de demonstração pelo que uma vez executado o utilizador passa de imediato e em exclusivo a observador.

```
10 REM OVAIS
20 MODE 1:CLS:CLG
30 DEG
40 X=320:Y=200:R=50
50 ORIGIN X,Y
60 FOR A=1 TO 360
70 PLOT R*COS(A),R*2*SIN(A)
80 NEXT A
```

```
10 REM TROCA DE COR EM CIRCULOS
20 MODE 0:CLS:CLG
30 FOR N=1 TO 15:INK N,INT(RND*27):NEXT
N
40 X=INT(RND*639):Y=INT(RND*399):R=INT(R
ND*80)
50 CI=INT(RND*15)
60 BORDER INT(RND*27)
70 IF R=0 THEN 30
80 S=R*R
90 R=R-1
100 IF R=0 THEN 30
110 Z=SQR(S-R*R)
120 L=Z+Z
130 SL=X-Z
140 IF SL<0 THEN L=L+SL:SL=0
150 PLOT SL,Y+R,CI:DRAWR L,O,CI
160 IF Y-R<0 THEN 90
170 PLOT SL,Y-R,CI:DRAWR L,O,CI
180 GOTO 90
```

A FEBRE DO OURO

NESTE jogo terá que guiar uma cobra através do écran utilizando o joystick, ou um conjunto de teclas escolhidas pelo utilizador. O objectivo consiste em apanhar todos os sacos de ouro amarelos e, se possível, as taças brancas que aparecem de vez em quando.

Nunca, mas nunca mesmo, deverá tentar apanhar uma taça vermelha.

Por cada saco amarelo que apanhar receberá 10 pontos e por cada taça branca 100 pontos. Se conseguir um bom resultado será admitido no quadro de "high scores". Se, sem dúvida por um mero caso de sorte, alguma vez conseguir chegar ao écran 15, verá que, a partir daí, as coisas se complicam bastante. O programa é relativamente pequeno e a separação das linhas dá ideia de como funciona. Partindo do principio que, mesmo assim, considera o programa demasidamente extenso, e percebe um pouco de programação em BASIC, pode tentar reduzi-lo um pouco substituindo algumas funções CHR$ por códigos de controlo introduzidos directamente.

Não tente correr o programa antes de o ter introduzido completamente e de o ter armazenado em suporte magnético, isto porque a chamada ao endereço BB03H que o programa utiliza, não limpa só o buffer do teclado, mas desactiva também, parcialmente, a tecla **ESC**ape.

A única forma de parar o programa, uma vez executado, é esperar até que apareça o quadro de "High-Scores" e, ignorando as duas opções apresentadas - **C** para continuar e **O** para opções de teclado - premir a tecla **E** (**E**xit).

60-220	-	Rotina principal. Lê o estado do teclado ou o joystick, verifica eventuais colisões, imprime a cobra, e decide quando deve apresentar as taças.
230-240	-	Aumenta a classificação e produz um som quando é recolhida alguma taça, aumentando ainda o comprimento da cobra.
250-260	-	Aumenta a classificação e produz um som quando é apanhado algum saco de ouro, aumentando simultâneamente o comprimento da cobra.
270-280	-	Produz som e reduz vidas quando se toca na parede, na cobra, ou taça vermelha.
290	-	Retira a taça do ecrã depois de um certo espaço de tempo.
300-310	-	Imprime uma taça de uma cor ao acaso em posição aleatória dentro do écran.
320-500	-	Rotina de "High Scores".
510	-	Diminui o período de tempo disponivel para jogar, e verifica o fim do jogo.
520-630	-	Imprime o ecrã.
640	-	Rotina para impressão de colunas.
650-780	-	Rotina encarregue das opções de joystick, ou teclado, que permite a utilização de teclas definidas pelo utilizador.
790-820	-	Cria caracteres definidos pelo utilizador.
850	-	Define variáveis, envelopes, e dimensiona matrizes para a posição da cauda.
860-960	-	Trata dos novos "High Scores".
970	-	Define funções para verificar as posições do écran.

```
10 INK 1,24:INK 0,1:INK 2,20:INK 3,6
20 RANDOMIZE TIME:CALL &BB03
30 GOSUB 790
40 GOSUB 650
50 CLS:GOTO 410
60 IF INT (RND*100)>98 AND FG=1 THEN GOS
UB 300
70 IF FG=0 THEN COU=COU-1:IF COU=0 THEN
GOSUB 290
80 IF WI<1 THEN SOUND 131,0,50,7,0,0,3:F
RED=REMAIN (1):VIDAS=VIDAS-1:IF VIDAS>0
THEN 540 ELSE GOTO 320
90 LOCATE A,B:PEN 1:PRINT CHR$(240):PRIN
T CHR$(22);CHR$(1):LOCATE A,B:PEN 3:PRIN
T CHR$(241):PEN 1:PRINT CHR$(22);CHR$(0)
100 IF E>0 THEN LOCATE C(E),D(E):PRINT "
 "
110 IF E>180 THEN E=0
120 E=E+1
130 IF F>180 THEN F=0
140 F=F+1:C(F)=A:D(F)=B
150 X$=UPPER$(INKEY$):IF X$="" THEN X$=Z
$
160 IF X$=L$ THEN T=FN T1:Z$=L$:A=A-1:GO
TO 210
170 IF X$=R$ THEN T=FN TR:Z$=R$:A=A+1:GO
TO 210
180 IF X$=U$ THEN T=FN TU:Z$=U$:B=B-1:GO
TO 210
190 IF X$=D$ THEN T=FN TD:Z$=D$:B=B+1:GO
TO 210
200 GOTO 60
210 IF T=0 GOTO 60 ELSE ON T GOSUB 230,2
50,270
220 ON FF GOTO 540,320,60
230 IF T=1 THEN FG1=0:SOUND 4,0,10,5,0,0
,8:FOR N=1 TO 3:F=F+1:IF F>181 THEN F=1
240 C(F)=A:D(F)=B:NEXT:SC=SC+100:LOCATE
#1,10,1:PRINT#1,USING"######";SC:FF=3:RE
TURN
250 IF T=2 THEN SOUND 4,0,10,5,0,0,8:FOR
 N=1 TO 3:F=F+1:IF F>181 THEN F=1
260 C(F)=A:D(F)=B:NEXT:SC=SC+10:LOCATE #
1,10,1:PRINT#1,USING"######";SC:HOM=HOM+
1:IF HOM=20 THEN SH=SH+1:FRED=REMAIN (1)
:FF=1:RETURN
 ELSE FF=3:RETURN
270 IF T=3 THEN SOUND 131,0,50,7,0,0,3:F
RED=REMAIN (1):VIDAS=VIDAS-1:IF VIDAS>0
THEN FF=1 ELSE FF=2
280 RETURN
290 FG=1:IF FG1=1 THEN LOCATE AA1,BB1:PR
INT" ";:RETURN ELSE RETURN
300 AA1=INT (RND*37)+2:BB1=INT(RND*18)+2
:IF FN CH=2 OR FN CH=3 THEN RETURN ELSE
SOUND 1,30,0,0,1:LOCATE AA1,BB1:IF RND<0
.5 THEN PEN
1 ELSE PEN 3
310 PRINT CHR$(245):FG=0:FG1=1:COU=100:R
ETURN
320 MODE 0:IF SC>VAL(MID$(A$(1),13)) THE
N PEN 15:PRINT"O SEU SCORE ESTA";TAB(7);
CHR$(10);"ENTRE OS DEZ 1os" ELSE GOTO 41
0
330 LOCATE 6,6:PEN 13:PRINT"NOME ?";CHR$
(10);TAB(3)"(MAX 10 LETRAS)":LOCATE 6,10
:PRINT STRING$(13," "):LOCATE 6,10:CALL
&BB03:INPUT
"",N$:N$=UPPER$(N$)
340 IF LEN(N$)>10 OR LEN(N$) <1 THEN 330
350 CHECK=10
360 WHILE SC<VAL(MID$(A$(CHECK),13))
370 CHECK=CHECK-1
380 WEND
390 IF CHECK>1 THEN FOR N=1 TO CHECK-1:A
$(N)=A$(N+1):NEXT
400 A$(CHECK)=N$+STRING$(11-(LEN(N$))+(7
-LEN(STR$(SC)))," ")+STR$(SC)
410 CLS:PEN 15:PRINT TAB(5)"HIGH  SCORES
";CHR$(10):FOR N=10 TO 1 STEP -1
420 PEN INT(RND*4)+1:PRINT TAB(2) A$(N);
CHR$(10);
430 NEXT
440 CALL &BB03
450 LOCATE 1,24:PEN 12:PRINT"PRIMA ";:PE
N 11:PRINT"C ";:PEN 12:PRINT"=CONTINUAR"
:LOCATE 1,25:PRINT"OU ";:PEN 11:PRINT"
 O ";:PEN 12
:PRINT "=OPCOES ";
460 SC=0:SH=0:VIDAS=3
470 IF INKEY(62)<>-1 GOTO 520
480 IF INKEY(34)<>-1 THEN GOSUB 650:CLS:
GOTO 410
490 IF INKEY(58)<>-1 THEN INK 0,13:INK 1
,0:PAPER 0:PEN 1:BORDER 13:MODE 2:LIST
500 GOTO 470
510 DI:WI=WI-1:PLOT WI,4,1:DRAWR 0,8,1:I
F WI=100 THEN SOUND 2,150,0,0,2:EI:RETUR
N ELSE EI:RETURN
520 INK 2,0:INK 0,0:INK 1,0:INK 3,0:BORD
ER 0:PAPER 0:MODE 1:WINDOW 1,40,1,21:WIN
DOW #1,1,40,23,23:WINDOW#2,2,39,2,20:PAP
ER #1,0
530 CLS:PEN 3:PAPER 0:FOR N=2 TO 20:LOCA
TE 1,N:PRINT CHR$(254);:LOCATE 40,N:PRIN
T CHR$(247);:NEXT:LOCATE 1,1:PRINT CHR$(
249);STRING$
(38,CHR$(248));CHR$(250);:LOCATE 1,21:PR
INT CHR$(251);STRING$(38,CHR$(253));CHR$
(252);
540 INK 2,0:INK 0,0:INK 1,0:INK 3,0:CLS#
2:PEN 3:PAPER 1:LOCATE 2,6:PRINT STRING$
((SH MOD 6)*4,CHR$(244)):LOCATE 40-((SH
MOD 6)*4),15
:PRINT STRING$((SH MOD 6)*4,CHR$(244))
550 IF SH>5 THEN PEN 3:PAPER 0:LOCATE 30
,2:GOSUB 640:LOCATE 10,15:GOSUB 640:IF S
H>11 THEN LOCATE 35,2:GOSUB 640:LOCATE 5
,15:GOSUB 64
0:IF SH>17 THEN LOCATE 25,2:GOSUB 640:LO
CATE 15,15:GOSUB 640
560 PAPER 0:LOCATE 10,5:PEN 1:PRINT CHR$
(240):PRINT CHR$(22)+CHR$(1):LOCATE 10,5
:PEN 3:PRINT CHR$(241)
570 FOR V=1 TO 20
580 AA1=INT (RND*37)+2:BB1=INT(RND*18)+2
:IF FN CH=2 OR FN CH=3 GOTO 580
590 LOCATE AA1,BB1:PEN 2:PRINT CHR$(242)
:LOCATE AA1,BB1:PEN 3:PRINT CHR$(243):NE
XT:PRINT CHR$(22)+CHR$(0):ORIGIN 0,0,0,1
00,12,4:CLG
3:ORIGIN 0,0,101,640,12,4:CLG 2:ORIGIN 0
,0,0,640,400,0
600 PEN#1,1:CLS #1:PRINT#1,TAB(4)"SCORE"
:LOCATE #1,10,1:PRINT#1,USING "######";S
C;:LOCATE#1,30,1:PRINT#1,"VIDAS ";VIDAS:
```

```
INK 2,24:INK
 0,0:INK 1,26:INK 3,6
610 HOM=0:Z=0:Z$="":E=0:F=0:A=10:B=5:C(1
)=10:D(1)=5:FG=1:WI=639:EVERY 4,1 GOSUB
510
620 CALL &BB03
630 GOTO 60
640 PRINT CL1$;:PRINT C12$;:PRINT C12$;:
PRINT C12$;:PRINT C12$;:PRINT C13$:RETUR
N
650 MODE 0:PAPER 13:CLS:BORDER 22:PEN 5:
LOCATE 7,6:PRINT"KEYBOARD":PRINT:PRINT T
AB(10)"OU":PRINT:PRINT TAB(7)"JOYSTICK":
PRINT:PRINT
TAB(8)"(K..J)"
660 CALL &BB03
670 IF INKEY(37)<>-1 THEN DU$=INKEY$:GOT
O 700
680 IF INKEY(45)<>-1 THEN I$=CHR$(8):R$=
CHR$(7):S$=CHR$(11):D$=CHR$(10):GOTO 780
690 GOTO 670
700 CLS:PEN 6:PRINT "DEFINA O TECLADO":P
RINT TAB(5)"TAL COMO E PEDIDO":PEN 5:LOC
ATE 6,7:PRINT"ESQUERDA:";:CALL &BB03
710 L$=UPPER$(INKEY$):IF L$="" GOTO 710
ELSE PRINT L$
720 PEN 7:LOCATE 6,9:PRINT"DIREITA :";:C
ALL &BB03
730 R$=UPPER$(INKEY$):IF R$="" OR R$=L$
GOTO 730 ELSE PRINT R$
740 PEN 3:LOCATE 6,11:PRINT"CIMA     :";:
CALL &BB03
750 U$=UPPER$(INKEY$):IF U$="" OR U$=L$
OR U$=R$ GOTO 750 ELSE PRINT U$
760 PEN 8:LOCATE 6,13:PRINT"BAIXO    :";:
CALL &BB03
770 D$=UPPER$(INKEY$):IF D$="" OR D$=L$
OR D$=R$ OR D$=U$ GOTO 770 ELSE PRINT D$
780 FOR N=1 TO 200:NEXT:INK 0,0:INK 1,24
:BORDER 0:PAPER 0:RETURN
790 SYMBOL AFTER 199:SYMBOL 200,255,192,
127,63,26,26,26,26:SYMBOL 201,255,3,254,
252,88,88,88,88:SYMBOL 202,26,26,26,26,6
3,127,192,25
5:SYMBOL 203,88,88,88,88,252,254,3,255:S
YMBOL 204,26,26,26,26,26,26,26,26:SYMBOL
 205,88,88,88,88,88,88,88,88
800 SYMBOL 249,255,180,205,182,157,210,1
72,187:SYMBOL 250,255,37,203,187,213,95,
169,149:SYMBOL 251,169,149,250,171,221,2
11,164,255:S
YMBOL 252,221,53,75,185,108,179,45,255:S
YMBOL 253,0,40,102,218,89,213,117,255
810 SYMBOL 254,168,252,198,248,154,228,1
40,240:SYMBOL 248,255,117,213,89,218,102
,40,0:SYMBOL 247,21,63,99,31,89,39,49,15
:SYMBOL 242,
0,126,24,60,126,126,126,60:SYMBOL 243,0,
0,29,2,0,0,0,0
820 SYMBOL 240,0,24,60,102,102,60,24,0:S
YMBOL 241,0,0,0,24,24,0,0,0:SYMBOL 244,2
38,238,0,187,187,0,238,238:SYMBOL 245,12
6,126,126,12
6,60,24,24,126:SYMBOL 246,0,42,84,42,0,0
,0,0
830 C11$=CHR$(200)+CHR$(201)+CHR$(8)+CHR
$(10):C12$=CHR$(204)+CHR$(205)+CHR$(8)+C
HR$(8)+CHR$(10):C13$=CHR$(202)+CHR$(203)
840 REM
850 DEFINT A-R,T-Z:DIM C(181),D(181):SC=
0:SH=0:VIDAS=3:ENV 2,127,6,1:ENV 1,1,15,
1,14,-1,5,5,0,1
860 DIM A$(10)
870 A$(10)="TRUNFAS       5000"
880 A$(9)="BRIOLANJO      4500"
890 A$(8)="EFISFERIO      4000"
900 A$(7)="ALIMPIO        3500"
910 A$(6)="TARQUINIO      3000"
920 A$(5)="BISTENCIO      2500"
930 A$(4)="CINFLINA       2000"
940 A$(3)="ESTRAGOSO      1500"
950 A$(2)="ANTRAQUIO      1000"
960 A$(1)="PAUPERIA        500"
970 DEF FN TR=TEST((A*16)+7,((25-B)*16)+
8):DEF FN T1=TEST((A*16)-25,((25-B)*16)+
8):DEF FN TU=TEST ((A*16)-9,((25-B)*16)+
24):DEF FN T
D=TEST ((A*16)-9,((25-B)*16)-8):DEF FN C
H=TEST ((AA1*16)-7,((25-BB1)*16)+8)
980 RETURN
```

Uma "spriteadela" no CPC 464

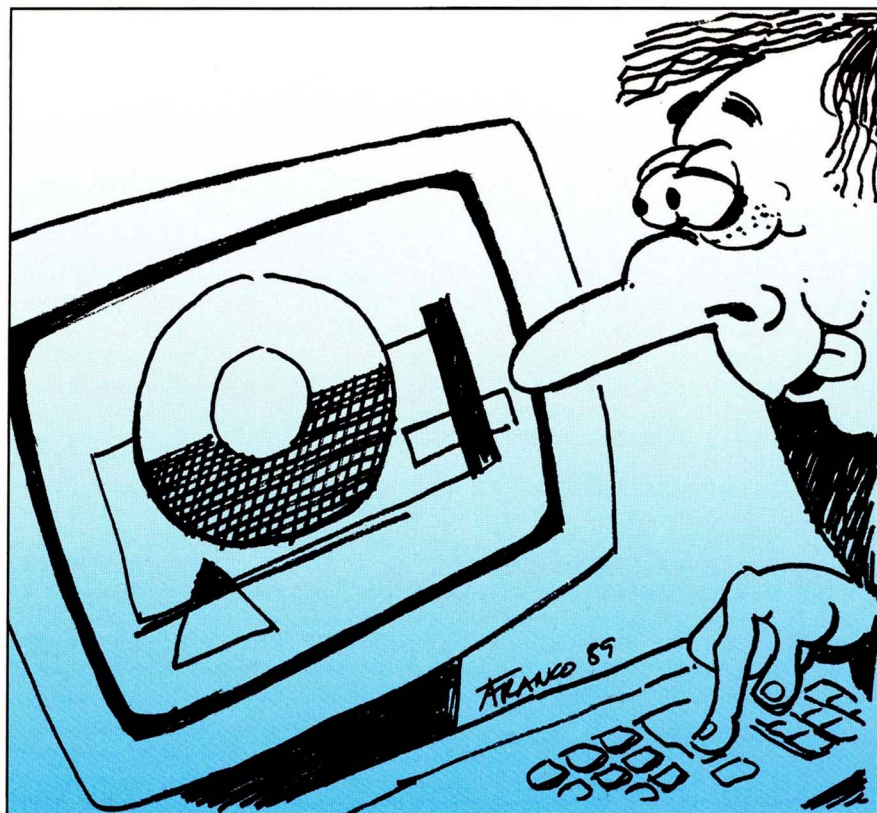
O CPC 464, apesar de mais limitado do que os seus sucessores, é um computador muito bem concebido e extremamente poderoso em todas as áreas com excepção de uma única: a dos gráficos.

Com efeito o **464** apenas dispõe das vulgares instruções **PLOT** e **DRAW**, ignorando pura e simplesmente os sprites, as instruções **FILL,** ou instruções **SCROLL**, tão comuns em muitas máquinas da sua geração, e em quase todas as que lhe foram posteriores. Este artigo tem o intuito de rectificar a deficiência que apontámos, acrescentando cinco novas instruções ao **BASIC** do **CPC 464**. Na realidade, trata-se de acrescentar mesmo novas instruções ao **BASIC** através de algumas rotinas em assembler, e não de aproveitar pequenas sequências de **POKE´s** e **CALL´s**.

A listagem 1, quando introduzida e executada, tornará disponiveis estas instruções, porém, antes de qualquer uma delas poder ser utilizada, tem de se activar o novo conjunto de instruções através de **CALL 37000**. As instruções agora adicionadas são as seguintes:

SPRITEON, X, Y, Z - Só funciona em modo 0. **X** é a coordenada do eixo **X** (valor situado entre 0 e 143), **Y** é a coordenada do eixo **Y** (valor situado entre 0 e 183), e **Z** o número do sprite (1 a 15). Se algum destes valores não se encontrar dentro dos limites acima especificados, a rotina devolverá o controlo ao interpretador de **BASIC**, sem ter efectuado qualquer tarefa.

SPRITEOFF,X,Y - Funciona em moldes idênticos aos da instrução **SPRITEON** retirando um sprite do ecrã em vez de o imprimir. Não é necessário indicar o número do sprite a retirar.

FILL,X,Y,Z - **X** é a coordenada do eixo **X**, e **Y** a coordenada do eixo **Y** (utilizando ambas as coordenadas os valores já referidos em **SPRITEON**), indicando o valor **Z** a cor em que o preenchimento da área especificada deve ser efectuado. Esta instrução funciona em qualquer modo de video.

SCROLLU - Executa um scroll de uma linha para cima.

SCROLLD - Executa um scroll de uma linha para baixo.

Todas as novas instruções agora adicionadas necessitam de ter como perfixo o caracter "|", comum aos comandos **RSX**, e conseguido através de **SHIFT**+"@". Este é o método que o programa interprete inclui para dizer ao **CPC** que os comandos novos existem sobre a forma de Extensões Residentes do Sistema.

Depois de ter introduzido a listagem 1 no computador e de a ter executado correctamente, guarde o programa-máquina resultante através de **SAVE "*comandos*", b, 37000, 500** e, sempre que necessitar destes comandos, carregue-os em memória através de **36999: LOAD "": CALL 37000**.

A segunda listagem apresentada (listagem 2), é um editor de sprites que cria

os códigos necessários para que estes posteriormente se possam reconstituir. Sem ele os dois primeiros comandos seriam quase inúteis.

Assim, para concluir a sua nova package de gráficos conseguida em estilo de "faça você mesmo", introduza e execute a listagem 2, e crie tantos sprites quantos desejar. Para esse efeito ser-lhe-à apresentada uma grelha com uma lista de cores de um lado e uma lista de comandos no topo. O cursor que pode ser movido na grelha - com o joystick 1, ou com as teclas de cursor - pode fixar ou apagar a cor, utilizando, respectivamente, a cor de tinta, ou cor de fundo, através da tecla "**COPY**", ou do botão de disparo. Desenhar com este editor é relativamente simples: ao premir a tecla "**COPY**" ou ou botão de disparo do joystick, a célula ocupada pelo cursor aparece preenchida pela cor de tinta activa, voltando novamente à antiga cor de fundo depois de premida a mesma tecla, ou botão. Os comandos disponiveis no topo da grelha são os seguintes:

P - Mudar de caneta (**P**EN) - as cores das canetas estão impressas à direita do écran.

I - Mudar a cor da tinta de uma caneta.

N - Começar o trabalho - ou continuá-lo - num outro sprite.

C - Apaga um sprite e utiliza uma grelha limpa.

S - Armazena os sprites em cassete.

Utilizando todos estes comandos podemos desenhar e gravar em cassete inúmeros sprites, por mais complexos que estes possam ser. Posteriormente todos eles podem ser carregados em memória através de **LOAD** "", e utilizados pelas instruções **SPRITEON** e **SPRITEOFF**, tal como referimos no inicio do artigo.

A última listagem deste artigo (a listagem 3), é uma simples demonstração das instruções criadas. Para observar a demonstração, introduza a listagem em causa carregando de seguida as novas instruções em memória, através de **36999: LOAD "" : CALL 37000**, e execute-a normalmente.

Esta demonstração explica como usar todas as novas instruções. Contudo, a parte mais interessante é, sem dúvida, a demonstração das instruções directamente ligadas aos sprites.

Utilizando uma das interrupções do **Z80**, as rotinas usadas conseguem movimentar o sprite no ecrã a uma velocidade constante, movendo-o um pouco e trazendo-o de novo ao lugar, conseguindo com isso o efeito de animação que permite, por exemplo, conceber "jogos de naves" em **BASIC** com uma velocidade satisfatória. Se, contudo, forem utilizados vários sprites e/ou forem deslocadas grandes porções de memória, o movimento torna-se irregular, podendo mesmo tornar-se muito pouco estético - ver a primeira demonstração de deslocação de sprites. Por outro lado, se se utilizar um número limitado de sprites e uma deslocação de, por exemplo, 1 pixel de cada vez, o movimento é extraordinariamente suave - ver a segunda demonstração de movimento de sprites.

Para finalizar deve referir-se que os comandos **FILL** e **SPRITEON** pressupõem uma cor de fundo em caneta 0, sendo imprevisível o que poderá acontecer numa outra situação, em todo o caso isto não deve constituir problema de maior dado que, ao ligar o computador este utiliza imediatamente uma cor da caneta 0 como cor do papel.

LISTAGEM 1

```
10 MEMORY 36999
20 CLS:PRINT "Pokeando, nao perturbe !"
30 X=37000:RESTORE
40 READ A$:IF A$="9999" THEN GOTO 80
50 FOR A=1 TO LEN (A$) STEP 2:POKE X,VAL
("&"+MID$(A$,A,2))
60 X=X+1:NEXT A
70 GOTO 40
80 CLS:PRINT "Terminado"
90 DATA 01929021C890CDD1BCC9A390C3D090C3
5091C35B91C34592C3519253505249544 54FCE53
50524954
100 DATA 454F46C646494CCC5343524F4C4CD55
343524F4C4CC4000000009290000000000DD7E00D
D4602DD4E
110 DATA 04FE00C8FE10D03D6778FEB9D079FE9
1D07C878787876F117C9219E53EC7906F5997675
779E60120
120 DATA 1ACD1DBC545DE13E10F5D5010800EDB
0D1EBCD26BCEBF13D20EFC9CD1DBC545DE1D5118
00019D13E
130 DATA 10F5D51AE6AA4F7EE655B1122313010
700EDB01AE6554F7EE6AAB11223D1EBCD26BCEBF
13D20DAC9
140 DATA 3E0FDD4600DD4E02C3E090DD7E00FE1
0D04FDD6603DD6E027CFE0238047DFE90D0DD560
5DD5E047A
150 DATA FE0338047BFE80D079F5
160 DATA CD11BCFE0020020E04FE0120020E02F
E0220020E010600CDE1BB322B92F1CDDEBBE5CDC
491CDE991
170 DATA CD139228F5E1CD2D922008CDC491CDE
99118F33A2B92CDDEBBC9E5D5E5D5C5CDF0BBC1D
1E1FE0020
180 DATA 13E5D5C5CDEABBC1D1E1EBA7ED42EB7
AFEFF20E0D1E1C9E5D5EB09EB7AFE0220057BFE8
03018E5D5
190 DATA C5CDF0BBC1D1E1FE00200BE5D5C5CDE
ABBC1D1E118DBD1E1C92B2B7CFEFF280CE5D5C5C
DF0BBC1D1
200 DATA E1FE00C93E03FE04C9010023237CFE0
120057DFE8F30EDE5D5C5CDF0BBC1D1E1FE00C90
601CD99BB
210 DATA CD2CBCCD4DBCC9060018F2000000000
0000000000000000000000000000000000000000
000000000
220 DATA 0000000000000000000000
230 DATA 9999
```

```
10 MODE 1:INK 1,13:INK 0,1:PAPER 0:PEN 1
20 MEMORY 36999
30 PRINT"Desenhador de sprites AMSTRAD M
AGAZINE":LOCATE 1,10
40 INPUT "Quantos sprites (1-15) ",SPR
50 IF SPR<>INT(ABS(SPR)) OR SPR>15 OR SP
R<1 THEN 40
60 SPR=SPR-1:DIM SP%(SPR,15,15)
70 KEY DEF 72,1,&F0,&F4,&F8
80 KEY DEF 75,1,&F3,&F7,&FB
90 KEY DEF 73,1,&F1,&F5,&F9
100 KEY DEF 74,1,&F2,&F6,&FA
110 KEY DEF 76,1,&E0,&E0,&E0
120 TSP=0:PN=1:XP=0:YP=0:GOSUB 550
130 GOSUB 490:GOSUB 140:GOTO 130
140 REM SCAN DO TECLADO
150 IF INKEY(62)=0 THEN 280
160 IF INKEY(27)=0 THEN 320
170 IF INKEY(60)=0 THEN 360
180 IF INKEY(46)=0 THEN 400
190 IF INKEY(35)=0 THEN 440
200 A$=INKEY$:IF A$="" THEN RETURN
210 IF A$<>CHR$(&E0) THEN 250
220 IF SP%(TSP,XP,YP)=0 THEN SP%(TSP,XP,
YP)=PN:GOTO 240
230 SP%(TSP,XP,YP)=0
240 GOSUB 520:RETURN
250 X=XP+(A$=CHR$(&F2))-(A$=CHR$(&F3)):Y
=YP+(A$=CHR$(&F1))-(A$=CHR$(&F0))
260 IF X>15 OR Y>15 OR X<0 OR Y<0 THEN R
ETURN
270 GOSUB 520:XP=X:YP=Y:RETURN
280 REM CLEAR
290 A$=INKEY$
300 CLS#1:INPUT #1,"Confirma (s ou n) ";
A$:IF A$<>"s" THEN GOSUB 770:RETURN
310 FOR A=0 TO 15:FOR B=0 TO 15:SP%(TSP,
A,B)=0:NEXT B:NEXT A:GOSUB 550:GOSUB 740
:RETURN
320 REM CANETA
330 A$=INKEY$
340 CLS#1:INPUT#1,"Qual a cor";A:IF A>15
 OR A<0 OR A<>INT(ABS(A)) THEN 340
350 PN=A:GOSUB 770:LOCATE 8,8:PAPER PN:P
RINT" ":PAPER O:RETURN
360 REM SAVE
370 GOSUB 780
380 SAVE"SPRITECODE",B,37500,4352
390 RETURN
400 REM SPRITE
410 A$=INKEY$
420 CLS#1:INPUT #1,"SPRITE ";A:A=A-1:IF
A>SPR OR A<0 OR A<>INT(ABS(A)) THEN 400
430 TSP=A:GOSUB 550:GOSUB 740:RETURN
440 REM INK
450 A$=INKEY$
460 CLS#1:INPUT#1,"Cor ";A:INPUT #1,"Nov
a cor ";B
470 IF A>15 OR A<0 OR A<>INT(ABS(A)) OR
B>26 OR B<0 OR B<>INT(ABS(B)) THEN 460
480 INK A,B:GOSUB 770:RETURN
490 REM PC
500 PLOT XP*16+4,YP*16+2,1:DRAWR 8,12:PL
OT XP*16+12,YP*16+2:DRAWR -8,12
510 RETURN
520 REM PS
530 FOR A=4 TO 12 STEP 4:PLOT A+XP*16,YP
*16+2,SP%(TSP,XP,YP):DRAWR 0,12:NEXT A
540 PLOT 304+4*XP,272+2*YP:RETURN
550 REM ECRAN
560 MODE 0
570 FOR A=0 TO 256 STEP 16
580 PLOT A,0,1:DRAWR 0,256
590 PLOT 0,A:DRAWR 256,0
600 NEXT A
610 FOR A=0 TO 15:LOCATE 14,A+9
620 PRINT A:PAPER A:LOCATE 18,A+9:PRINT"
 ":PAPER 0
630 NEXT A
640 LOCATE 15,7:PRINT"Cores"
650 PLOT 639,0:DRAW 639,316
660 DRAW 416,316:DRAW 416,0
670 DRAW 639,0
680 WINDOW#1,2,19,2,4
690 PLOT 0,399:DRAW 639,399:DRAW 639,320
700 DRAW 0,320:DRAW 0,399
710 GOSUB 770
720 LOCATE 1,7:PRINT"Sprite";TSP+1:PRINT
"Caneta ";:PAPER PN:PRINT" ":PAPER 0
730 RETURN
740 X=XP:Y=YP:FOR XP=0 TO 15:FOR YP=0 TO
 15:IF SP%(TSP,XP,YP) THEN GOSUB 520
750 NEXT YP:NEXT XP
760 XP=X:YP=Y:RETURN
770 PRINT#1,"P Caneta   C ClearS Grava",
"N Sprite   I Tinta":RETURN
780 'COMPILE
790 FOR A=0 TO SPR:CLS:PRINT:PRINT:PRINT
:PRINT"COMPILACAO EM":PRINT"PROGRESSO"
800 FOR B=0 TO 15:FOR C=0 TO 15:PLOT 4*B
,368+2*C,SP%(A,B,C):PLOT 4*B+101,368+2*C
,SP%(A,B,C):NEXT C:NEXT B
810 FOR C=0 TO 7:FOR B=0 TO 7:POKE 37500
+272*A+B+8*C,PEEK(49512+B+2048*C):NEXT B
:NEXT C
820 FOR C=0 TO 7:FOR B=0 TO 7:POKE 37500
+272*A+B+8*(C+8),PEEK(49232+B+2048*C):NE
XT B:NEXT C
830 FOR C=0 TO 7:FOR B=0 TO 8:POKE 37628
+272*A+B+9*C,PEEK(49164+B+2048*C):NEXT B
:NEXT C
840 FOR C=0 TO 7:FOR B=0 TO 8:POKE 37628
+272*A+B+9*(C+8),PEEK(49244+B+2048*C):NE
XT B:NEXT C
850 NEXT A
860 FOR A=(SPR+1)*272+37500 TO 41852:POK
E A,0:NEXT A
870 RETURN
```

```
10 MEMORY 36999:CALL 37000
20 X=37500:PRINT"ESPERE UM MOMENTO..."
30 READ A$:IF A$="9999" THEN 210
40 FOR A=1 TO LEN(A$) STEP 2:POKE X,VAL(
"&"+MID$(A$,A,2)):X=X+1:NEXT
50 GOTO 30
60 DATA 0044CCCCCCCC880044993333333336688
9933333333333336699323031323031669933330B1
723033
70 DATA 993332B17231336644333333333333388
0099333333333660000449933336688000000004433
33880000
80 DATA 0044993333668800009933669933 6600
4433669C6C9933889933883C3C44336699660014
28009966
90 DATA CC8800000000044CC0000CCCCCCCCCC00
0000CC3333333333CC0044333333333333338844
33303033
100 DATA 303033884433327033B031338844333
37033B03333880099333333333336600004433333
333338800
110 DATA 0000CC333333CC0000000000009933660
000000000CC333333CC000000443333CC3333880
0009933CC
120 DATA 3CCC3366004433 66143C28993388443
388003C0044338844CC0000000000CC88
130 DATA 000000050A00000000000000F0F00000
00000050F0F0A0000000000055AA50A0000000000005
AA5000000
140 DATA 00000050A0000000000000050A000000
000000050A000000000F0F0F0F0F00050CCCCC
CCCDC26A0
150 DATA E4CCCCCCCCB94CD8E4CCCCCCDC26CCD
850CCCCCCB94CCCA000F0F0F0F0F0000030000
000003000
160 DATA 10002000001000200000000000F00000
0000000000050F0A0000000000000F0F0F0000000
000000FF0
170 DATA 0F000000000000005F00A00000000000
000F0000000000000000F0000000000000000F
00000000
180 DATA 0050F0F0F0F0F0A00000E4CCCCCCCCB
9580050CCCCCCCCDC26CCA050CCCCCCCCB94CCCA
000E4CCCC
190 DATA DC26CCD8000050F0F0F0F0F0A000001
0200000000102000000201000000000201000
200 DATA 9999
210 FOR A=41580 TO 41851:POKE A,0:NEXT A
220 MODE 0:A=2:B=40:C=1:P=2:Q=100:R=2:X=
2:Y=140:Z=3
230 PRINT"SPRITES IRREGULARES"
240 EVERY 3,3 GOSUB 260
250 FOR T=0 TO 2000:NEXT T:GOTO 310
260 DI:¦SPRITEOFF,X,Y:X=X+Z:¦SPRITEON,X,
Y,1:¦SPRITEOFF,P,Q:P=P+R:¦SPRITEON,P,Q,2
270 ¦SPRITEOFF,A,B:A=A+C:¦SPRITEON,A,B,1
:IF A=142 OR A=1 THEN C=C-C
280 IF P=142 OR P=0 THEN R=-R
290 IF X=143 OR X=2 THEN Z=-Z
300 EI:RETURN
310 Z=REMAIN(3):DI:CLS:R=1:Z=1:Y=180:Q=6
0:PRINT"SPRITES REGULARES":EI
320 FOR A=0 TO 600:CALL &BD19:GOSUB 390:
NEXT A
330 CLS:PRINT"FILL & SCROLL":FOR T=0 TO
5000:NEXT T:CLS
340 FOR A=0 TO 640 STEP 40:PLOT A,0:DRAW
R 0,400,1:NEXT
350 PLOT 0,0:DRAW 636,0,1:DRAW 636,398,1
:DRAW 0,398,1:DRAW 0,0,1
360 FOR A=0 TO 15:¦FILL,A*40+20,200,A:NE
XT
370 FOR A=1 TO 24:FOR B=0 TO A:¦SCROLLU:
NEXT B:FOR B=0 TO A:¦SCROLLD:NEXT B:NEXT
 A
380 STOP
390 DI:¦SPRITEOFF,X,Y:X=X+Z:¦SPRITEON,X,
Y,1:IF X=0 OR X=142 THEN Z=-Z
400 ¦SPRITEOFF,P,Q:P=P+R:¦SPRITEON,P,Q,2
:IF P=0 OR P=142 THEN R=-R
410 EI:RETURN
```

IMPRESSÃO A DUAS COLUNAS

Construir textos a duas colunas é fácil com este pequeno programa em BASIC. Aqui vai, portanto, o "truque" para os fãs do LocoScript.

CONSTRUIR um texto a duas colunas sempre foi uma necessidade de alguns dos nossos leitores. Foi pensando em resolver essa dificuldade que decidimos proporcionar neste artigo uma solução para o caso através do BASIC. Este programa pega num documento feito no LocoScript e imprime-o a duas colunas duma só vez. Mas faz mais - sai tudo perfeitamente alinhado (coisa que era difícil com outros métodos) e chega mesmo a imprimir no fundo da página o seu número correcto.

A listagem só funciona em ficheiros ASCII, de maneira que temos de perder todos aqueles comandos muitos cómodos do LocoScript. Também não se pode esperar tudo de uma listagem BASIC de meia centena de linhas. Digite a listagem e grave-a em disco ou fita antes de preparar um documento no LocoScript.

É uma boa ideia começar por criar um TEMPLATE.STD para um novo grupo LocoScript, de maneira a manter nele toda a "escrita" a duas colunas. Assim, crie um documento chamado TEMPLATE.STD e prima duas vezes a tecla [f7]. O menu que vai aparecer-lhe vai questioná-lo sobre o tamanho das páginas. Responda-lhe que quer o tamanho de página 66, a zona do cabeçalho (Header) em 0 :Position I e a zona de rodapé (Footer) em II :Position 62. Isto deve dar-lhe um corpo de página de 55. Faça o EXIT deste menu e prima [f1] para definir o aspecto geral da página (Layout). A seguir defina as margens para 0 e 37 e o espaçamento (pitch) para 12, ficando correcto o template.

Escreva então o seu texto, mas sem atributos como, por exemplo, letras e palavras em negrito (bold) ou sublinhadas. E não se esqueça de que a última linha do texto não pode ficar em branco ou vazia. Grave finalmente o texto produzido, e use o [f7] para criar um ficheiro ASCII ("Create an ASCII file"), sem se esquecer de escolher a opção "Page Image". Grave o resultado final no grupo 0 (o CP/M, e consequentemente o BASIC, só funcionam normalmente com ficheiros integrados neste grupo).

Carregue agora o BASIC e execute o programa que introduziu previamente. Ele irá pedir-lhe o nome do ficheiro do LocoScript e o número de linhas de página (entre 1 e 55). Chegado a este ponto tem a possibilidade de imprimir o texto imediatamente, ou vê-lo primeiro no ecrã para ver se ficou com bom aspecto e sem erros (Use [Alt]+S para parar e/ou continuar com o scroll). O programa está preparado para imprimir com o tipo de caracter Draft Elite, mas você pode sempre comutar para Near Letter Quality, como de costume, através de [PTR].

```
10 CLEAR : ESC$=CHR$(27) : CLS$=ESC$+"E"+ESC$+"H"
20 LPRINT ESC$;"@";ESC$;"M";ESC$;"d"
30 PRINT CLS$ ; "PRINT DE IMAGEM DE UMA PAGINA DE UM FICHEIRO ASCII EM DUAS COLUNAS"
40 PRINT : INPUT"Ficheiro = ",NAME$
50 IF FIND(NAME$)="" THEN PRINT "Nao encontro. -Tente de novo":GOTO 40
60 PRINT:INPUT"Maximo de linhas por pagina. (1 to 55) = ",NUM
70 IF NUM < 1 OR NUM > 55 THEN 60
80 PRINT : PRINT"Carregue em V  para ver no ecra ou P para imprimir"
90 SELECT$=UPPER$(INKEY$):IF SELECT$="" THEN 90
100 IF SELECT$ = "V" OR SELECT$ = "P" THEN 110 ELSE 90
110 PRINT CLS$0! PRINT"Abrindo o ficheiro :-",NAME$ : PRINT
120 IF SELECT$="P" THEN GOSUB 400
130 :
140 LINES=0 : PAGE=1 : OPEN "I",1,NAME$
150 DIM A$(110)
160 WHILE NOT EOF(1) AND LINES < NUM*2
170 LINES=LINES-1
180 LINE INPUT #1,A$(LINES)
190 IF LEFT$(A$(LINES),1)=CHR$(12) THEN GOSUB 430
200 WEND
210 LINES=LINES-1
220 IF LINES < (NUM*2) THEN GOSUB 460
230 IF NUM=0 THEN CLOSE 1 : ERASE A$.: GOSUB 410 : GOTO 330
240 FOR X=1 TO NUM
250 PRINT TAB(5)A$(X);TAB(45)A$(X+NUM)
260 NEXT X
270 IF SELECT$="V" THEN PRINT:GOTO 290
280 FOR K=1 TO 57-NUM:PRINT:NEXT K
290 PRINT TAB(41-(PAGE < 10))PAGE; CHR$(12)
300 LINES=0 : PAGE=PAGE+1 : ERASE A$
310 GOTO 150
320 :
330 PRINT"Prima R para reutilizar o mesmo ficheiroe, N para um ficheiro novo, ou F para fim"
340 ZZ$=UPPER$(INKEY$):IF ZZ$="" THEN 340
350 IF ZZ$="R" THEN PRINT CLS$ : PRINT"Ficheiro = ",NAME$ : GOTO 60
360 IF ZZ$="N" THEN 30
370 IF ZZ$="F" THEN PRINT CLS$:NEW
380 GOTO 340
390 :
400 POKE 8793,234 : RETURN
410 POKE 8793,239 : RETURN
420 :
430 C%=LEN(A$(LINES))-1
440 A$(LINES)=RIGHT$(A$(LINES),C%): RETURN
450 :
460 NUM=INT(LINES/2)
470 IF LINES/2 <> NUM THEN NUM=NUM+1
480 RETURN
```

PCW8256 = VT52?

INTRODUÇÃO

Desde que comprei o meu AT compatível questiono-me frequentemente acerca do destino a dar ao meu velho PCW8256. Quando me convidaram para fazer este trabalho tive a agradável surpresa de constatar que não me ia limitar apenas a servir-me dele como máquina de escrever. Descobri que o poderia ligar ao meu AT e usá-lo como terminal sobre o sistema operativo XENIX. E isso porque o PCW8256 possui um programa de emulação, o MAIL232. O que não possui é a porta série RS232 que terei de adquirir. O seu preço? Não sei, mas de certeza é compensatório à compra de um novo terminal.

Mas vejamos pormenorizadamente o que pode fazer com o seu PCW8256.

HARDWARE DO PCW8256

Em termos de Hardware, o PCW8256 (abreviação de Personal Computer Word Processor) vem com um monitor, um teclado e uma impressora.

Até agora o computador parece completo, mas para que possa comunicar com outros computadores necessita ainda do RS232C (que é comercializado separadamente) e de um cabo série.

Apesar de tudo, em termos de Software, para além de encontrarmos no PCW um processador de texto bastante jeitosos e do sistema operativo CP/M (potente, mas talvez desactualizado), podemos encontrar ainda o programa de comunicações MAIL232.

O QUE É O MAIL232?

O MAIL232 é o software que lhe permite facilmente ligar o seu PCW a outro, e, através da definição de alguns parâmetros (tais como Ratios de Transmissão, Bits de controlo e de informação, paridade, etc.), transferir ficheiros entre eles ou ainda fazer com que o PCW lhe faça a emulação de um terminal muito semelhante a um Zenith Z19/Z29, o que faz dele um dos muitos terminais a emular o VT52 de maneira que pode servir de terminal a outros sistemas tais como VAX/VMS ou ainda XENIX ou mesmo UNIX.

Mas então, aonde está esse programa? Pode encontrá-lo facilmente na sua disquete do processador de texto, escondido na directoria do sistema. Como se chega lá? Vamos ver: depois de fazer o boot com a disquete do sistema, vire-a e dê o comando DIRSYS para obter os ficheiros contidos na directoria do sistema. Repare bem e lá encontrará o ficheiro MAIL232.COM que poderá chamar facilmente bastando para isso escrever na prompt:

```
A>MAIL232
```

Se tudo estiver a corre bem surge-lhe a mensagem:

```
Amstrad PCW8256 Electronic Mail Terminal
 Requires RS232C / Centronics Interface
 ®1985 Amstrad Consumer Electronics plc.
```

Depois disto chega finalmente ao menu do programa:

```
F1=Framing ▌  F3=Files ▌  F5=Connect ▌  F7=OFF ▌  MAIL TERMINAL PROGRAM V1.2
```

Poderá usar as teclas F1, F3, F5 ou ainda F7 para chamar os menus que desejar, como lhe é indicado.

O COMANDO DEVICE DO CP/M

Ainda com o PCW, mas já sem o programa MAIL232, você poderá compreender melhor as comunicações entre o hardware se estudar o comando DEVICE. Com este você poderá efectuar as ligações entre os seus canais lógicos (nomes lógicos atribuídos pleo CP/M) e os canais físicos (os da saída para a impressora e/ou para o ecrã, para a entrada pelo teclado, etc.), e ainda alterar parâmetros de comunicações associados, tal como, por exemplo, o protocolo a usar.

Mas observemos melhor este comando...

O CP/M Plus suporta os seguintes devices lógicos:

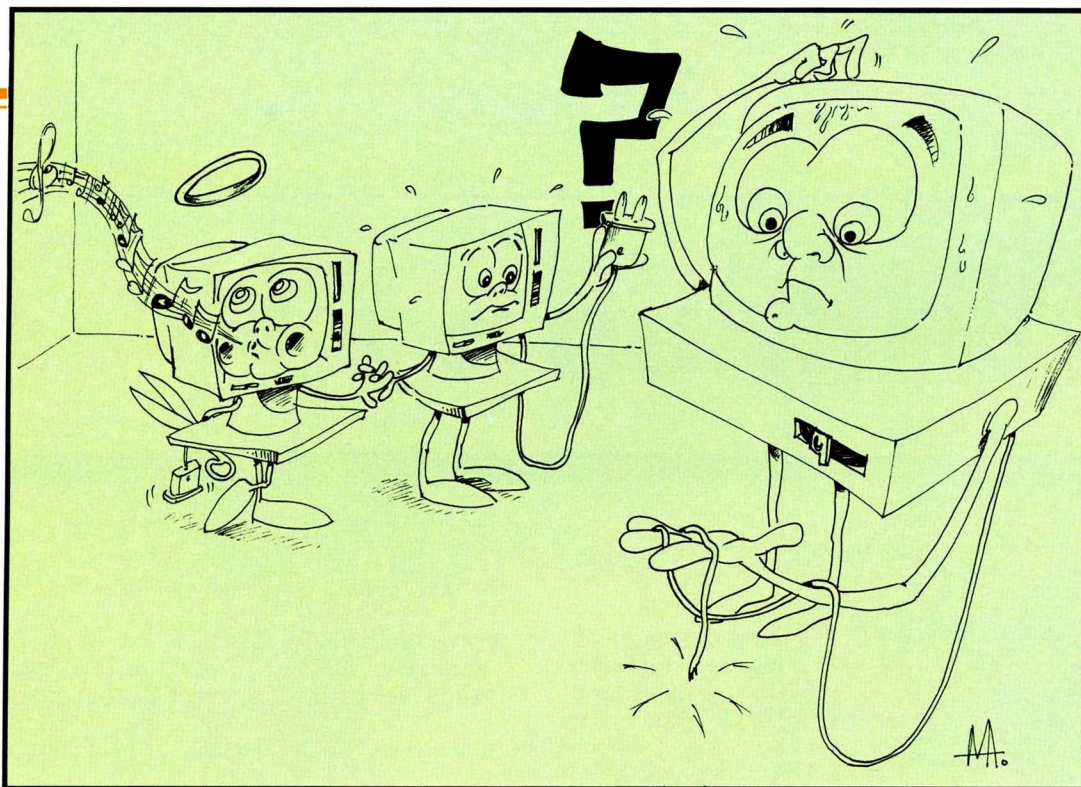
COMIN: entrada pelo teclado
CONOUT: saída pelo ecrã
AUXIN: entrada de um device auxiliar
AUXOUT: saída de um device auxiliar
LST: a impressora

Os devices físicos dependem da configuração do seu hardware. Por exemplo, CRT (Cathode-Ray Tube) designa o monitor.

As suas diferentes formas de uso (a sua sintaxe) permitem a verificação dos estados correntes e/ou fazer as desejadas alterações.

DEVICE { **NAMES** | **VALUES** | dev-físico | dev-lógico }
DEVICE dev-lógico = dev-físico { opção } {, dev-físico {opção}, ... }
DEVICE dev-físico { opção }
DEVICE CONSOLE [**PAGE** | **COLUMNS** = colunas | **LINES** = linhas]

Mas vejamos cada caso detalhadamente:

No primeiro caso poderemos obter os devices físicos e as atribuições correntes dos devices lógicos (DEVICE), uma lista dos devices físicos com uma descrição sumária das

características de cada um (DEVICE NAMES), as atribuições de cada device lógico (DEVICE VALUES), ou os atríbutos de um device físico ou lógico (por exemplo, DEVICE CON).

No segundo caso podemos atribuir um device lógico a um device físico, por exemplo, DEVICE CONOUT:=LPT,CTR que dirige o output para a impressora e para o ecrã.

No terceiro caso poderemos desconectar a ligação anteriormente estabelecida entre o device lógico e o respectivo device físico, igualando o primeiro a NUL (nulo). O último caso permite-nos modificar o status da consola. Um exemplo será alterar o número de linhas ou colunas que vão aparecer no ecrã.

NOVAMENTE O MAIL232

Aquilo que se passa no MAIL232 é praticamente o reflexo do que acontece com o comando DEVICE, para a comunicação de dois computadores, havendo ainda que estabelecer mais alguns parâmetros. Senão vejamos:

No primeiro menu (aquele que é accionado pela tecla <f1>), você poderá definir a velocidade de transmissão (Tx Baude Rate) e de recepção (Rx Baud Rate) entre os dois computadores — as suas opções variam entre 50 e 19200 Baud, dependendo do Hardware que se pretende pôr a comunicar — o número de bits a transmitir para suporte da informação (que em CP/M são 8, em XENIX são 7, etc.), o tipo de paridade a usar (paridade par — quando a soma do número de bits no estado ON (1) é par e o bit de paridade vem a OFF (0) quando a informação é correctamente transmitida — paridade impar — quando a soma do número de bits no estado ON é impar e o bit de paridade vem a OFF (0) quando a informação é correctamente transmitida; ou ainda nenhum tipo de paridade); e o número necessário de bits a transmitir para controlar a transmissão (Stop bits).

F1=Framing | F3=Files | F5=Connect | F7=OFF

Tx Baud Rate 9600
Rx Baud Rate 9600
Data bits 8
Parity .... NONE
Stop bits 1
H/W Handshaking OFF

No menu accionado pela tecla <f2> poderá indicar qual o ficheiro a transmitir e/ou o ficheiro a receber.

F1=Framing | F3=Files | F5=Connect | F7=OFF

Receive [,]
Send [ ]
Transfer as ASCII

No seguinte menu (tecla <f5>) poderá estabelecer o seu estado na linha (ONLINE — correctamente a transmitir ou a receber; ou LOCAL — estabelecendo parâmetros nos outros menus sem estar a transmitir ou a receber).

F1=Framing | F3=Files | F5=Connect | F7=OFF

ONLINE / local

Por último, o menu da tecla <f7> estabelece o modo de emulação Z19/VT52 ou a saída para o CP/M.

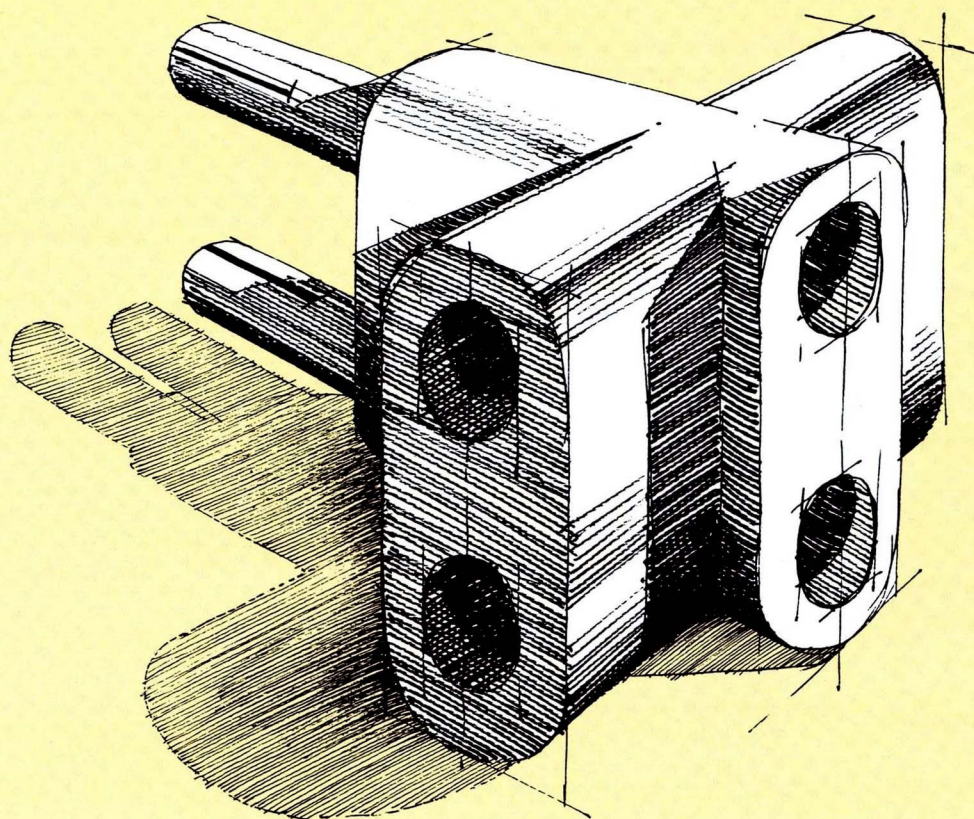
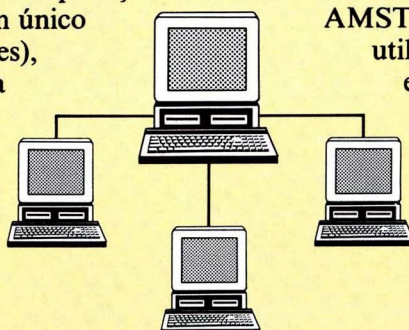
F1=Framing | F3=Files | F5=Connect | F7=OFF

Z19 / VT52 Emulation
* RETURN TO CP/M +*

É então muito simples, como se pode ver, colocar o PCW a comunicar sem grandes complicações bastando para isso ter um RS232C.

CONCLUSÕES

Para quem já tenha esta máquina, pode até nem ser má ideia usá-la no âmbito do Office Automation, porque embora um pouco lenta (4,77/8MHz) pode executar facilmente três funções: a de processador de texto (que executa perfeitamente), a de correio, e a de terminal (emulando o VT52) de um sistema de maior porte usando a facilidade trazida pelo MAIL232.

P.P.

COMPRO/VENDO/TROCO

COMPRO

Amstrad 1512. 2ª mão. Contactar Paulo Neves, Horta dos Fumeiros, 14 - 1º Dto, 8000 Faro.

Jogo Xadrez p/ 15M/ PC envie instruções p/ Luís Alberto 2125 Couço.

Jogos p/ CPC 464 em cas. 500$00 cada contactar Cristóvão Morgado, Estrada de Pataias nº 22 Burinhosa 2445 Pataias.

Compro monitor Neptun 156 avariado. Preço a combinar. Victor Santos, Estrada do Paço 8365 Algoz.

Qualquer modelo Amstrad 1512 - 1640 em bom estado. Miguel Duque Tel. 98212317 a qualquer hora e dia.

Esquecemos PC1640 DOS 3 Monitores e unidade de sistema. Adalberto G. Cabete, apartado 178, 3000 Coimbra.

PC 1640 HDM+IMP-DMP 3000, em bom estado. Contactar Tl. 053/26205 ou R. Simões de Almeida, 19 - 4700 Braga — Claudia Domingues.

Software e modem para RTTY e CW em PC compatível. Resposta para: C. Soares. Rua da Alegria 785-AP 21-4000 Porto.

Procuro compilador C para PCW 8512 Amstrad Telefone 386131/11 das 9h às 12/13h e das 14h às 18h para Manuel Mealha.

VENDO

Jogos e utilitários p/ Spectrum e Spectrum +. Requisite Lista de programas/preços. Pr. 1º Maio, 13 - 3880 Ovar.

Vasta Gama de software IBM-PC a baixo preço. Pedir lista para apartado 4361, 1508 Lisboa codex.

Vendo Sinclair QL com 4 utilitários + 4 Cartriges por 35c, contactar Miguel Correia, Tel. 7153473 Lisboa/Benfica.

Vendo interface para joystic e também transformador de corrente. Tudo para Spectrum. Escassa utilização. Tel. 21881 Chaves. Alexandre Teixeira.

Amstrad CPC664 com vários programas, cabo para impressora, cabo para gravador, monitor monocromático, tudo por 40c. Tel. 2765116.

Amstrad 1512 DDMM por 150 contos, ocm 7,5 Mb de software, impecável. António Russo, Rua José Relvas, 347 - 351, 2090 Alpiarça.

Amstrad 6218 (cores) + cabos impre. e gravador + jogos, 90c. Tel. 9811243 Odivelas. Contactar Rui ou Paulo, depois das 21h39.

ZX Spectrum+2, manual + 50 jogos + joystick com pouco uso por 38 contos. Contactar M. Santos Ribeiro. Abelheira 3720 O. Azemeis, tel. (056) 64787.

Casio FX-750P, nova + 2 livros com programas, contactar 053/26205 Claudia Domingues.

Hyundai-At 640KB/20 Mb 1Dr. 5 1/4 Placa Quadram - CGA - Ega Monoc. 6 slots 102 Tec. Imp. Citizen + rato 350c. PP Fernando, Tel. 2245728.

Programa de Medicina útil para Cardiologia, é simples de usar, é rápido, não é necessário tempo de aprendizagem. Preço 3 550$00.

Monitor Amstrad ECD novo. Contactar Amândio Silva pelo Tel. (039) 715792 a partir das 19h00.

Curso de microcomputador MSDOS básico ou avançado ministrado casa do cliente; intensivo ou pós-laboral. Marcação

Tel. (01) 4101561.

TC 2068 Timex = 18 500$00, monitor = 8 500$00, impressora TC 2080 = 41 000$00.

Fechadura para o seu micro/PC-bloqueamento total e garantido no acesso à informação do disco rígido. Contactar Tel. (01) 4101561.

Vendo Amstrad PC 1512 com duas drives, com ou sem impressora. Estado novo. Tel. 310646.

ZX 48K, interface 1+Micro Drive, 2 gravadores, joystick c/ interface, (1 cartridge + 60 jogos + livros = oferta) tudo 50c. Tel. 075-46419.

Computador + gravador MSX monitor Philips e cassetes 50c. António Magalhães. Rua Augusto Moreira. V. N. Paiva.

Spectrum 128K 2+ em bom estado. Contactar Pedro Castel-Branco. Tel. 4101024 das 19 às 21h. 25 mil escudos.

Amstrad PC 1512 HD20 MM impecável baixo preço com ou sem impressora e secretária com rebaixo teclado Dust Covers muitos programas.

Programas para calculadoras Casio. Inversão de matrizes, determinantes, sistemas, estatística, etc. Mais informações. Tel. 788776 (01).

100 revistas sinclair user - computer + vídeo games - your sinclair - crash-Zx computing - computer gamer. Tel. 901264 Lisboa.

Software PCW 8512 com software do PC1640, aceito propostas por favor telefonar para F. Gomes, 4747149 depois das 21 horas.

Amstrad CPC 6128, 128 K jogos, programas, joystick. Bom preço, possível troca. Falar com Paulo Monge, depois das 8 horas. Tel. 02-813393.

Executo programas para PC e compatíveis, bom preço. Entrega rápida. Hora de jantar falar com Luís Mendes. Tel. 02-683132 Porto.

Detector de radar radatec, capta frequ. 9-11,50 GHz pela melhor oferta. João Pedro Beito - Lama - V. N. de Muia, 4980 Ponte da Barca.

TC2048 com menos de 1 ano de utilização e leitor de cassetes e algum software por 20c. Pedro - Abrantes. Tel. (041) 23255.

Vendo 2068 com gravador + monitor + garantia Timex. Preço 45c. Rua José Falcão 144 Porto. Tel. 24717 falar João F.

Jogos para Amstrad1512. Para mais detalhes e catálogo, enviar 100$00 para: Computer Dream (c) Apartado 19, Valbom 4421 Gondomar.

Calculadora TI 57 LCD programable + manual de instruções, estado novo 5c. Rui Oliveira Inácio (049) 312329.

Turbo Pascal para CP/M mais Cobol supercalc2 Masterfile 8000, com os manuais e as disquetes, tudo por 30c. Tel. 4747149.

Vendo jogos para PC's, CPC, Spectrum, MSX, muitos e baratos, contactar Inforams, Av. Elisyo de Moura, 397 - 2º - 3000 Coimbra.

PPC 640 Amstrad - 2 drives e monitor monocromático Philips de 14" novos, preço 220c. Aceito troca por equip. com disco. Tel. 4373496.

Newbrain + disc 200K + Seikosha GP100A manuais cabos, etc., ou troco. Tel. (02) 9711100 ou R.

COMPRO/VENDO/TROCO

Ribeiro Teles, 462 Ermesinde.

Executo programas do seu interesse. Contacte João Saraiva. R. S. Bento das Peras, 122 - 2, 2º - 4435 Rio Tinto. Tel. 9892898 Porto.

Vendo software para o Commodore 64, Spectrum e Amstrad PC1512. Escreva para: Ricardo P., Rua das Rosas, 36 r/c Esq., Moreira Mais 4470.

PC Amstrad 1512 Double Drive e impressora Amstrad DMP 3160 mais software: Ability, Sympony, WS, DA, Jogos, etc. por 250c.

Amstrad CPC464 64K monocromático, estado novo 50c. Tel. (052) 633531 SA. Dom. 4480 Vila do Conde.

Criamos software de aplicação para PC's em 5 1/2 e 3 1/2. Sonosoft. R. Rocha Peixoto 16 r/c 4490 Póvoa de Varzim.

Transformo o seu PC 1512 em PC 1640 e coloco filtros em ecrãs. Contactar Miguel Beirão. Tel. 480857 Porto-

Executo qualquer tipo de programas em DBase. Telefone 811759 Porto.

Calculadora Casio FX 750 P(7), programável em Basic, cientifíca 4KB de memória, expansível a 16 KB, com manual. Tel. 4312038, Lisboa.

Printer Amstrad LQ3500 como nova, Tel. 4373263 (à noite).

Amstrad CPC464, monocromático + Floppy disc drive 3" (DD1) + software. Como novo. Nuno Veríssimo. Tel. 832461 Lisboa.

Vendo Amstrad DMP3160. Pouco utilizada em bom estado. Vendo por 40c. contactar em Lisboa Tel. 820536.

Computador Spectrum 48K/10 jogos grátis. Meio ano, pouco uso por apenas 12c. Nilton Areal 4785 Trofa Tel. 44520.

Spectrum+3, monitor, joystick, gravador, disquetes, livros, cabos, CP/M, cassetes, tudo isto por 90c. Tel. 9214398, Viais ou Fenando.

Software por medida, Pascal, DBase III, Basic - todo tipo de software. Contactar: Luís Madaleno, Tel. 2592260 -14/24H.

Spectrum +2, 28c. urgente copio todo o tipo de software de 5 1/4 p/ 3 1/2 e vice-versa. Mário Encarnação. Ap. 38 - 5000 Vila Real.

Comp. TC 2048, monit. verde, imp. papel term., Grav. Sanyo DR100, joystick, sint. voz, I3000 porg. conj. 65c., Ind. Suj. oferta. A. Santos 7630 Zambujeira.

Amstrad CPC 664 + monitor GT65 + impressora DMP1. com manuais e embalagens. Pouco uso, óptimo estado. 70c. Tel. 01-8145634.

Ventura publisher em português, paradox, etc. Jogos Atari St. prog. PCW, etc. Inforsystem, Apartado 125/89 - 2670 Loures.

Amstrad CPC6128, drive incorporado, monitor mono, disquetes com utilitários e jogos. Trata 568932 (Paulo).

Spectrum 48K c/ monitor verde/ preto c/ cabos e cassetes. Vendo por 30 contos. Contactar jorge. Tel. 4748356.

Programa original de cálculo de sapatas, p/a Engenheiros civis Demodisk e adicionais, mais informações, Tel. (01) 2592260 (14 - 24h.).

FDD 3000 + disquetes + livros. Em conjunto ou separado. Tel. 693475 (Lisboa).

Computador Amstrad CPC464 c/ monitor. Utilitários (Hisoft Pascal e Hisoft Devpac) c/instruções. Alguns jogos. Tel. 4941763 Amadora.

Monitor CD para 1512 ou 1640, preço a combinar. Contactar: Ricardo Silva, Tel. (062) 23497 aos fins de semana.

Monitor poli./média resol. p/ Amstrad 1640, como novo. Preço a combinar. Contactar Ricardo Silva, Tel. 062-23497 (Sab./ Dom.).

Monitor policromático, média resolução, da Amstrad. Preço a combinar. Contactar: Ricardo Silva, Tel. (062) 23497. Fim de semana.

Spectrum 48k avariado + 3 livros + manual + cabos de ligação + 8 cassetes com programa 5c. Tel. 28246 Braga, depois 19.

QL 128 K, 30c., impressora A4 40c., monitor Philips 20c., conjunto 80c. António Russo, Tel. 043-26525 - 26549 H. Exped.

Mini comp. texas TI 74 + interface p/ gravador + cartridge 8K RAM por 25 contos. Tel. 283329.

Vende-se Casio FX730P memória RAM 16K, Interface para gravador FA5 25c. Tel. 2262369 (Lisboa) depois das 20h.

Atari 520 ST completamente novo, 1 Drive. Contactar António Mendes, Tel. 26841 Porto. (Dias uteis das 18 às 20 horas).

Sinclair QL 128K expansível a 512K, 2 micro drives 25c. + QL printer 100 CPS 45c., vários programas tudo 80c. Tel. 071-29265.

Sinclair QL 128 K expansível a 512K, 2 micros drives, vários programas, processador de texto, folha de cálculo, etc. 35c. Tel. 071-29265.

Impressora para sinclair QL. QL printer 100 CPS muito bom estado para uso 45c. Tel. 071-29265.

Segurança para PC com disquete chave não copiável bit a bit. Tel. (01) 4151125 (a partir das 8 h).

Amstrad 1640 DD HD como novo, 10 meses, por 180c. Forneço todo o tipo de software. Contactar Tel. 06332074 entre as 20 e 24h.

Spectrum 128K +2 com joystick + jogos por 34c. tudo está como novo contactar António Ferreira, Tel. 73082 (053) Braga 18-20h

Amstrad 1512 HD (1 ano) + impressora Brother Twinriter 5 inumero software (texto, dados, utilit), em conjunto ou Sep. Tel. 01/ 9261616.

Amstrad PCW 8512 + Locoscript 2 + protext + DBase II + Turbo Pascal + Xadrez. 130c. Tel. 656263 depois das 18h.

Faço p/ encomenda e vendo programas educativos compilados. Deseja catálogo escreva a: Francisco Peixoto. L. Resende Mire de Tibaes, 4700 Braga.

Vende-se Timex 2048K com gravador timex com cassete de instruções e livro (português) Tel. 54302, Pedreira 3220 Miranda do Corvo.

Vendo Spectrum +3 com vários extras, pouco uso, barato. Tel. 2904152.

Todo o tipo de software, envio à cobrança. Só programas completos e com manuais. Inform-System, Apartado 125/89 - 2670 Loures.

Software PC, PCW, Atari ST c/ manuais, peça as tabelas de software. Para o apartado 125/89 - 2670 Loures. Envia-nos pelo correio.

COMPRO/VENDO/TROCO

TROCO

Quero programa I'Ching, Poker Machine, Cad, Draw Poker, etc... Para PC 1640. Tel. 805880 de Lisboa, José Rui também vendo Foguetão.

Tenho Flight simulator 3, e Larry (the Land of the Louge Lizards) para trocar por Alex Higgins World Snooker. Tel. 082-21494.

Todo o tipo de software (principalmente jogos) para PC. Contactar Sérgio Henriques. tel. 7594200 Lisboa entre as 21h e as 22h.

Lista de software PC compatível. + 150 programas variados. Envie-me a sua lista. Pedro Lúcio. Al. Conde de Oeiras, 48-2780 Oeiras.

Jogos e utilitários para Amstrad CPC 6128 contactar: João Dinis, Rua Júlio dinis Nº91 - 1º Esq. Hab-12 - 4000 Porto.

Trocamos, compramos e vendemos todo tipo de software para PC's. Sonosoft, R. Casa Poveiros do Rio 657-12 Esq. 4490 Póvoa de Varzim.

Troco jogos e programas para o ZX Spectrum e Amstrad PC 1640. Tel. 805880 José Rui. Enviem lista ou telefonem - Lisboa.

Troco software p/ PC compatível enviar lista p/ troca. Filipe Ganança. R. Pedro da Fonseca nº12 B 2º - 6000 Castelo Branco.

Troco software utilitários p/ Spectrum 48K / Opus Discovery. Cont. José Eduardo Marques, Bairro Chasa, Lote 2, nº33 - 2615 Alverca.

Troco 6 disquetes 5.25 2HD BASF por 6 disquetes normais (estão novas, compreia-s por engano), contactar Mirandela Tel. (078) 23233.

Trocamos software p/ PC. Contactar Dnsoft — Restaurante "O Casarão" — Azóia 2400 Leiria. Tel. (044) 27980 ou (044) 34205 (só noite).

Amstrad 1512 DDMM por Atari 520 ou 1040, PPC, ou Amiga 500. António M. C. Russo. Rua José Relvas 347-351 2090 Alpiarça.

Software para PC 1512 e compatíveis. Troco lista, para trocas. Escreve para António B. Craveiro, J. de Freguesia 7600 Messejana.

Possuo um Amstrad PC, gostaria de trocar software e ideias com outros utilizadores. Contactar Alexandre Teixeira. Campo da Roda 5400 Chaves. Tel. 21881.

Troco todo o tipo de software para PC's. José M. Sagorro. Horta das Figueiras 66-a 3ºEsq. 7000 Évora. Tel. (066) 27206.

Troco ideias e software para o PCW com pessoas que vivem na zona de Lisboa. Rua dos Remédios 143 - 1º por escrito. Tel. 875903.

Software p/ PC — Técnico e recreativo — favor enviar lista para: C. Soares, Rua da Alegria 785 AP 21-4000 Porto.

Programas para PC's compatíveis. Grande variedade. Carlos Ladeira, Vila Franca do Rosário 2665 Malveiras.

Revistas e programas para PCW com outros utilizadores. Artur Tomé (01) 9881264 (depois das 20h00.

Troco Word, Windows, DBase 3 Plus, Turbocad, Works, Tur. Pascal, Lotus, Tas-Word, WS2000, WS1512, PFS, Pmaster, Nortonutil, PCTools e muitos jogos. António Maldonado, R. Josefa de Óbidos, nº22 r/c Dto. tel. 865527 a partir das 19h.

TRIUDUS
INFORMÁTICA
DIVISÃO PROFISSIONAL
soluções
equipamentos
formação

Hoje, existe o novo Amstrad PCW 9512.

Mais do que uma evolução natural na escrita, o novo AMSTRAD PCW 9512 é uma revolução.
Corrige, alinha frases, substitui palavras, personaliza cartas, a impressora faz cópias automaticamente... e, todos os textos ficam registados em arquivo numa diskette, prontos a serem utilizados.
O novo processador de texto AMSTRAD PCW 9512 executa todas estas tarefas com rapidez, simplicidade e eficiência. É mesmo revolucionário!
Não pense que o AMSTRAD PCW 9512 lhe vai custar mais do que uma máquina de escrever electrónica, lembre-se que AMSTRAD é qualidade a baixo preço. Ponha no lixo a sua incompetente e ultrapassada máquina de escrever.

Amstrad PCW 9512
A revolução na escrita.

Cominfor
COMPANHIA PORTUGUESA DE INFORMÁTICA

AMSTRAD